



PHD

**Challenges in artificial socio-cognitive systems: A study based on intelligent vehicles**

Baines, Vincent

*Award date:*  
2015

*Awarding institution:*  
University of Bath

[Link to publication](#)

**Alternative formats**

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (<https://creativecommons.org/licenses/by-nc-nd/4.0/>). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

**Take down policy**

If you consider content within Bath's Research Portal to be in breach of UK law, please contact: [openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk) with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.

# Challenges in artificial socio-cognitive systems: A study based on intelligent vehicles

submitted by

Vincent Baines

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Science

30/09/2014

## **COPYRIGHT**

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author .....

Vincent Baines

# Abstract

Technological developments are causing a proliferation of computing devices in every day life, with the availability of high compute power, small size, and the capability for wireless communication leading to consideration of how to off-load human tasks to devices which can manage themselves. Similarly, the science of how artificial systems can operate in an environment, sensing, reasoning, and taking action, has become increasingly mature. These developments lead to the opportunity for artificial entities to undertake activities in the real world, but facing significant challenges in reasoning about the environment and social interaction with humans.

The concept of artificial entities operating amongst humans raises a set of socio-cognitive problems, that is, issues where reasoning is required both about the environment and human activity, raising the challenge of the need to understand the cultural and contextual aspects of a situation. Further challenges stem from giving intelligent entities the autonomy to pursue their own goals. Firstly, how to manage the situation when, in simple terms, an entity does not know what to do, for example it has no appropriate knowledge of how to handle the situation it finds itself in, or enters into a conflict which it cannot resolve by itself. Secondly, we consider the challenge of how an entity's pursuit of its own goals can be balanced against the greater social welfare, where an entity may be taking action which is to the detriment of the wider population, or where coordination of its action could result in benefit for others. In broad terms, we consider these as issues relating to an entity's understanding of the environment in which it operates, and we adopt the concept of Situational Awareness as a means to analyse this understanding. We consider an entity's understanding as being built up from low level perceptions (where events in the environment are sensed) to an increased understanding at the comprehension level (where possible meanings of the perceptions are generated) through to a high level projection understanding (of the likely future state of the environment).

We refine these issues into a number of problem statements which we see artificial socio-cognitive systems (ASCS) facing, and propose the approach of attempting to build an explicit representation of SA at different levels, and how to move data, information and knowledge upwards through them. We then explore this by grounding experimentation in the domain of intelligent vehicles. This problem space contains a number of characteristics which make it suitable for our work: there is complex human interaction, rules which may not completely govern the situation or be adhered to, and technological developments in autonomous vehicles, all of which can be represented in scenarios in order to assess our approach. We use this domain

to illustrate our proposed approach, and rather than develop solutions for this specific domain we remain abstract as far as possible in order to be able to offer conclusions that can find application in other domains.

We describe a framework where distributed components are brought together to support investigation into these problem areas. A generalised message exchange approach is adopted, with messages containing additional semantic annotation such that the emphasis for appropriate handling lies with the consumer. Concerning *what* is exchanged, we consider knowledge and understanding in terms of Situational Awareness levels as a means of allowing components to communicate at an appropriate level and prevent communication overload due to exchanging the wrong kind and the wrong volume of data.

We present scenarios that have been constructed to illustrate problematic aspects in the domain that are reflections of the wider challenges faced by artificial socio-cognitive systems, and show how these can be tackled or mitigated with the help of a range of framework components. An intelligence layer contains autonomous agents which are responsible for controlling vehicles, but we assist these agents through the use of an external governance structure, capable of issuing guidance to the intelligent agents for situations where they do not know what to do, and/or to issue appropriate obligations to ensure the wider society goals are met. We see such external regulation as an intrinsic feature of social systems, whether implied (convention) or explicit (regulation/law), and replicate this structure through the use of institutions to provide a reference when the agent's knowledge is incomplete.

In conclusion, our focus is on agent understanding from a Situational Awareness perspective, and consideration of what communication at which level is appropriate; from this we find that agents are better suited to higher level communication than to dealing with the processing of high volumes of low level perceptions. We couple the intelligent agents to an external governance structure, to mimic existing structures of regulation allowing agents to be provided with additional guidance as required. This approach is demonstrated in a number of scenarios which show the framework resolving issues where an individual agent would otherwise show undesirable behaviour as it: i) lacked knowledge of social convention, ii) would choose to pursue its own benefit over the collectives, or iii) lacked appropriate jurisdiction over other agents to bring about a solution. Having demonstrated how these issues can be alleviated by our approach in particular scenarios, we argue for some generalisations to problems broadly faced by (artificial) socio-cognitive systems, which we believe have sufficiently similar characteristics that our approach to the concretisation of SA for ASCS can be applied.

# Acknowledgements

After this many years, it is difficult to look back and think exactly who to mention. So, first a covering statement, thanks to all the people who have offered advice, support, ideas, and encouragement along the way.

Huge thanks to the opensource community, and all those who give their great ideas and software development for others to benefit from. It would have been extremely difficult to produce this work without such tools being given freely for others to use.

Thanks to my supervisor, Julian Padget, for the support and guidance along the way. It's been a steep learning curve, but thanks for helping me find my voice.

Thanks to the family: my parents, sister, and the whole Hobbs family.

Finally, the most important thanks saved till the last, thanks to Rachael Baines.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Problem statements . . . . .	14
1.2	Objectives . . . . .	16
1.3	Contributions . . . . .	18
1.4	Thesis structure . . . . .	19
1.5	Related Publications . . . . .	20
<b>2</b>	<b>Related research</b>	<b>21</b>
2.1	Human factors . . . . .	22
2.1.1	Situational Awareness . . . . .	22
2.1.2	Knowledge sharing . . . . .	26
2.2	Computational representation . . . . .	32
2.2.1	Intelligent Systems . . . . .	32
2.2.2	Agent organisation . . . . .	38
2.2.3	Communication frameworks . . . . .	40
2.3	Problem Domain . . . . .	43
2.3.1	Intelligent Vehicles . . . . .	44
2.3.2	Vehicle communication . . . . .	46
2.3.3	Entity Simulation . . . . .	47
2.3.4	Traffic Simulations . . . . .	50
2.4	Summary . . . . .	52
<b>3</b>	<b>Analysis of the problem</b>	<b>54</b>
3.1	Generalisation of problem type . . . . .	54
3.1.1	Knowledge and understanding . . . . .	54
3.1.2	Introduction of novel technologies . . . . .	56
3.1.3	Ability of intelligent systems . . . . .	57
3.1.4	Constructing distributed systems . . . . .	58
3.2	Problem statements . . . . .	59
3.3	Defining problem in a specific domain . . . . .	63
3.4	Investigative system requirements . . . . .	65

3.4.1	Information exchange . . . . .	65
3.4.2	Intelligent control . . . . .	67
3.4.3	Governance . . . . .	68
3.4.4	Problem domain representation . . . . .	68
3.4.5	Review of results . . . . .	69
3.5	Summary . . . . .	70
<b>4</b>	<b>Developing an architecture and software solution</b>	<b>72</b>
4.1	Components and interfacing . . . . .	73
4.1.1	Messaging framework . . . . .	74
4.1.2	Intelligence layer . . . . .	77
4.1.3	Entity representation . . . . .	87
4.1.4	Institution Framework . . . . .	97
4.1.5	Area Of Interest . . . . .	101
4.1.6	RDF Utilities . . . . .	103
4.1.7	3D Viewer . . . . .	108
4.1.8	Final system . . . . .	110
4.2	Performance and testing . . . . .	110
4.3	Summary . . . . .	114
<b>5</b>	<b>Implementation of scenarios</b>	<b>115</b>
5.1	Building scenarios . . . . .	116
5.1.1	Developing initial behaviours . . . . .	116
5.1.2	City and Motorway . . . . .	117
5.1.3	Validation and testing . . . . .	118
5.2	Perception driven scenarios . . . . .	120
5.3	Waypoint exchange . . . . .	124
5.4	Projection of future events . . . . .	125
5.5	Resolving problematic situations . . . . .	130
5.6	Variable speed limits . . . . .	134
5.7	Post accident management . . . . .	139
5.8	Summary . . . . .	142
<b>6</b>	<b>Analysis of results</b>	<b>144</b>
6.1	Preliminary work . . . . .	145
6.1.1	Incorrect geospatial calculations . . . . .	145
6.1.2	Incorrect management of agent belief base . . . . .	146
6.1.3	Inappropriate data transmission rates . . . . .	147
6.2	Metrics . . . . .	148
6.2.1	Communication metrics . . . . .	148
6.2.2	Vehicle performance metrics . . . . .	149
6.2.3	Convoy performance metrics . . . . .	150

6.2.4	Motorway metrics . . . . .	150
6.3	Analysis of scenarios . . . . .	152
6.3.1	Perception driven . . . . .	152
6.3.2	Waypoint exchange . . . . .	158
6.3.3	Projection of future events . . . . .	161
6.3.4	Resolving unknown behaviours . . . . .	163
6.3.5	Variable speed limit . . . . .	166
6.3.6	Post-accident management . . . . .	171
6.4	General findings . . . . .	175
6.4.1	Knowledge and communication . . . . .	176
6.4.2	Introduction of novel technologies . . . . .	177
6.4.3	Ability of intelligent systems . . . . .	178
6.4.4	Constructing distributed systems . . . . .	180
6.5	Summary . . . . .	181
<b>7</b>	<b>Further Work</b>	<b>183</b>
7.1	Improving intelligent vehicle scenarios . . . . .	183
7.1.1	Quality of simulated and extracted data . . . . .	184
7.1.2	Variety of scenarios . . . . .	184
7.1.3	Validity of behaviours . . . . .	185
7.1.4	Remaining questions . . . . .	185
7.2	Exploring further applications . . . . .	186
7.2.1	Adding human in the loop capability . . . . .	187
7.2.2	Extending to alternative devices . . . . .	187
7.3	Further problem questions . . . . .	188
7.3.1	How to manage dynamic information exchange . . . . .	188
7.3.2	When directives are ignored or violated . . . . .	189
7.3.3	Time considerations . . . . .	189
7.4	Summary . . . . .	190
<b>8</b>	<b>Conclusions</b>	<b>192</b>
8.1	Findings against problems . . . . .	192
8.2	Closing remarks . . . . .	200
<b>A</b>	<b>Additional institution definitions</b>	<b>202</b>
A.1	Crash group institution . . . . .	202
A.2	Emergency services institution . . . . .	204
A.3	Insurer institution . . . . .	205
A.4	Motorway management institution . . . . .	206



# List of Figures

2-1	Müller-Lyer Illusion . . . . .	27
4-1	Reproduction of Wooldridge’s situated agents . . . . .	78
4-2	Integration of agent infrastructure to domain environment . . . . .	79
4-3	Tankcoders example view . . . . .	88
4-4	XMPP vehicles from phase one migration . . . . .	90
4-5	XMPP vehicles from phase two migration . . . . .	91
4-6	Android XMPP Vehicle . . . . .	92
4-7	AR Drone Pub-Sub rate . . . . .	94
4-8	Institution information flow . . . . .	98
4-9	Institution integration into the BSF . . . . .	99
4-10	Multiple institutions integrated into the BSF . . . . .	100
4-11	Area of Interest displayed in 3D viewer . . . . .	102
4-12	Example output of rdfTest tool components . . . . .	104
4-13	rdfMonitor network metrics . . . . .	106
4-14	rdfMonitor domain metrics . . . . .	107
4-15	Improved 3D xmppViewer tool . . . . .	109
4-16	BSF module deployment . . . . .	110
4-17	Messaging test hardware configuration . . . . .	112
4-18	Openfire vs ejabberd performance for RDF messages . . . . .	112
4-19	Openfire vs ejabberd performance for JSON messages . . . . .	113
5-1	Example of early success and failure . . . . .	116
5-2	Route waypoint verification (Map data ©2013 Google) . . . . .	119
5-3	Route alignment problem shown using xmppViewer component . . . . .	120
5-4	Vehicle convoy route around Bath city centre . . . . .	122
5-5	Vehicle route in Bath city centre . . . . .	127
5-6	Vehicle motorway crash scenario . . . . .	140
6-1	3D view augmented with collision volume data . . . . .	146
6-2	Case of convoy failure during data saturation . . . . .	147
6-3	Measurement of vehicle braking . . . . .	149

6-4	Measurement of convoy gaps and speeds . . . . .	151
6-5	Percept count over time . . . . .	153
6-6	Message count over time . . . . .	154
6-7	RDF Message count over time . . . . .	155
6-8	Deviation from lead vehicle route . . . . .	156
6-9	Distance to lead convoy vehicle . . . . .	157
6-10	Performance of perception based approach . . . . .	160
6-11	Performance of waypoint approach . . . . .	161
6-12	Fuel use comparison with and without institution active . . . . .	162
6-13	Institution states for traffic light management scenario . . . . .	162
6-14	Vehicle speed and gap comparison for individual lane . . . . .	165
6-15	Institution states for ‘flash lights’ resolution . . . . .	166
6-16	Agent controlled speed comparison for VSL scenarios . . . . .	167
6-17	Vehicle speed and gap comparison for VSL scenarios . . . . .	169
6-18	Institution states for VSL scenario . . . . .	170
6-19	Speed comparison per lane for all vehicles . . . . .	172
6-20	Agent vehicle speeds . . . . .	173
6-21	Institution states for crash management scenario . . . . .	174

# Listings

2.1	An RDF example structure and content . . . . .	41
4.1	An RDF annotation generated by the BSF test package . . . . .	75
4.2	Central agent info belief . . . . .	82
4.3	Driver agent tempInfo action . . . . .	82
4.4	Driver agent entityUpdate action . . . . .	82
4.5	Driver speed plan . . . . .	83
4.6	Driver distance-speed beliefs . . . . .	83
4.7	Driver agent arrived plan . . . . .	84
4.8	Driver move plan . . . . .	84
4.9	Convoy info plans . . . . .	85
4.10	Convoy management plan . . . . .	86
4.11	Convoy distance behaviours . . . . .	86
4.12	Convoy distance ahead plan . . . . .	86
4.13	Convoy waypoint plans . . . . .	87
5.1	Traffic lights institution definition . . . . .	129
5.2	Road users institution definition . . . . .	133
5.3	Variable Speed Limit institution definition . . . . .	138
6.1	Agent reaction to slowDown event . . . . .	167
A.1	Crash group institution definition . . . . .	202
A.2	Emergency services institution definition . . . . .	204
A.3	Insurer institution definition . . . . .	205
A.4	Motorway management institution definition . . . . .	206

# Chapter 1

## Introduction

Computing technology continues to become an ever growing part of human life, and although perhaps not the fully integrated vision of 20th century films and literature, it is hard to find a facet of existence not touched in some way. In many cultures, this is shifting from being a more passive one-way interaction (e.g. the computational power behind pharmaceutical research or weather predictions) to a more interactive exchange (e.g social media, augmented reality). Additionally, we are seeing intelligent devices becoming more established, from devices able to autonomously navigate and clean accommodation, to autonomous air vehicles capable of take off, flight to a specified destination, and landing.

There are certainly aspects of concern in such developments, with film and literature giving a (perhaps exaggerated) warning to numerous generations to tread carefully. It is difficult not to think of Arthur C. Clarke's HAL 9000 when considering what could go wrong in handing over control of decision making to an artificial intelligence system, even ignoring the fictional context of the book. Certainly most programmers have been confronted with the situation where their code has produced an unexpected result, and whilst resetting a block of code is hopefully problem free, dealing with a run away autonomous system caught in a control loop between sensors and actuators, raises more concern.

Drawing on HAL again, we see a reflection of the concern amongst humans in handing over responsibility to another (in this case, system). Understandably, the more critical the activity being transferred to the other, the more reassurance and safety precautions are sought (for example, an autopilot system, or life support machine). At the least, we require some mechanism where the system can be disengaged and control returned to the human operator (e.g. to avoid HAL's "I'm sorry, Dave, I'm afraid I can't do that" response). The ability for an autonomous system to somehow explain its decision making process, in order for the human to gain an insight into its understanding of the current situation, would be desirable. This would help allay the concern of the human(s) involved that the system has 'gone wrong' when in fact it is performing correctly, bridging the gap between the human understanding of the situation, and the systems'.

As such systems increasingly enter into day to day human life, the social, legal, and ethical

issues around their use become debated. Such topics were discussed by McCarthy [2009], which also put forward a summary of differences between system approaches (quoted here) as:

- Controlled systems: where humans have full or partial control, such as an ordinary car
- Supervised systems: which do what an operator has instructed, such as a programmed lathe or other industrial machinery
- Automatic systems: that carry out fixed functions without the intervention of an operator, such as an elevator
- Autonomous systems: that are adaptive, learn and can make ‘decisions’.

This provides a good starting point to differentiate between the available approaches to controlling an end device (e.g. a piece of machinery). Looking historically, there seems to be a trend that human operators are happy to cede *some* autonomy to another system (for example, to avoid the monotony of a simple repetitive task), but less willing to cede *all* control to a fully autonomous system.

In this thesis, we consider the adoption of near-future disruptive technology into human society, and which computing theories and approaches can support this. Whilst controlled, supervised and automatic systems will undoubtedly continue to serve a use, we posit that autonomous systems will become a more prevalent choice, allowing humans to offload decision making activities to ‘smarter’ systems.

Exploring the use of autonomous systems, in terms of areas of concern touched upon earlier, we can begin to identify some problem areas. There is a central issue around understanding: how does an autonomous system develop an understanding of what is happening in the environment in which it is operating? Relating to this, the problem of how it would communicate this understanding to others within the environment? Can the system better aid others if it communicates not just what it knows, but what its future plans and intentions are? Taking this further, how is the situation handled if another entity has conflicting beliefs to that which the autonomous is stating, is confidence lost? Or can the two views be ratified somehow? If humans are involved, how is the situation managed where the autonomous system has the correct understanding but the human does not (especially if the human loses confidence and over-rides the autonomous system)?

We draw on the topic of ‘Situational Awareness’ (SA) as described by Endsley [1995], as a means to consider various levels of information exchange and developing understanding. Although arguably lacking an exact specification, this provides a framework for considering information ‘levels’ at which components may wish to exchange (e.g. from simple perceptions, through to projection of the implication(s) of those perceptions) and also act upon. We broaden this topic of SA from its perhaps more common military domain usage, to a wider domain of application to technical innovations, i.e. as devices become more aware, how can we consider them in an SA context, and what benefits does this bring?

Considering the requirements around the autonomous system aspect of the problem, we adopt a Multi Agent System (MAS) approach, based on the Belief-Desires-Intention model put

forward by [Bratman et al. \[1988\]](#). This allows a collection of agents to be defined in a more abstract AI appropriate language, empowered with the ability to pursue given goals. This approach offers some solution to the observability problem put forward earlier, with agents able to provide a narrative explanation around belief updates and plan selection.

With a set of autonomous agents pursuing some goal(s), we are then faced with the question: What happens when they are unable achieve that goal or fulfil an obligation? This could be due to being prevented from achieving that goal in some fashion, for example another entity is using a required resource, or blocking in some fashion. It could also be due to the agent not having sufficient knowledge to achieve that goal, such as a goal of moving to a position when the agent has no route information. Even when an agent does have the ability to pursue a goal, there is an issue of how the balance between individual and social welfare should be addressed, i.e. an agents desire to focus on its own goals may be to the detriment of the wider society. For cases such as this, we first consider it in an abstract sense: that an agent knows ‘part of the picture’ of what is going on, and what its goal is, but is missing a ‘piece of the puzzle’ to fulfil its goal. In this situation, a possible approach could be to ask some other entity for guidance, but if this offers no assistance then we draw on the approach taken in human affairs, that such a case is referred to an external body which has appropriate jurisdiction to resolve the problem. An example of this could be receiving compensation from another entity over a breach of contract, where at an entity to entity level there is no enforcement mechanism to receive payment, but if the matter is handled by an appropriate institution (i.e. a criminal justice court), then the offending entity can be issued with an obligation to pay compensation.

For this aspect, we introduce the use of institutions as a system component, where the rules and process internal to the institution are defined in an abstract logically concise language, which when combined with the environment events results in (where appropriate) the institution issuing an obligation (e.g. in the justice example, providing a ruling). This provides us with a number of potential benefits, which in general terms we consider as a ‘steer’ of the agents autonomy, as whilst obligations should be obeyed, they do not necessarily have to be (although there should be some penalty for not doing so). This allows us to view the process more as a guide than a pure enforcement mechanism, and as such to broaden the application to include not just resolution of conflict, but wider coordination and assistance for situations where the agent requires such additional input. This has the benefit of reducing the effort involved in attempting to equip an agent with knowledge of how to handle every possible situation, instead being able to draw on guidance from the institution. To re-use the legal example, an agent isn’t required to carry the knowledge of every country’s legal process to cover all eventualities of operating in whatever country it finds itself in. Instead, it has the mechanism to refer the matter to the appropriate legal body for the region it is currently in, and allow that institution to provide the guidance for resolution. For intelligent agents operating socially within a community, we might consider the case where cultural or ambiguous events bring about a situation where an agent lacks understanding of what to do, and may even lack awareness that it does not know what to do. For example if I have never been told to return a bow greeting with a bow, I might not take any action in that cultural context. We consider this as an illustration for the

type of problem which we see socio-cognitive agents having to deal with, and we use the term socio-cognitive throughout this work to refer to agents which have to reason about both their physical environment as well as the social situation in which they operate. The example of the bow contains social elements of which the agent may be unaware such as duration and angle to bow, and we see a role for institutions in being able to offer assistance for cases such as this.

Following on from this, we address the practical application of the work: How can we construct a situationally aware intelligent multiagent system, with the ability to collaborate and resolve scenarios where the agent has no prepared response, but must work out what is appropriate either on its own or by recourse to external advice. In order to ground this question, a meaningful problem domain is required with which to construct test scenarios for the system to operate against. We are looking for a domain to operate in, where these problem characteristics can be investigated through the construction of appropriate scenarios. We are interested in cases where intelligent agents could be situated in some environment in which they have a task and appropriate stimulus to drive the intelligence of ‘what is going on’ and suitable environment-actions to bring about their ‘this is what I should do’. We seek a domain in which it is possible to represent human and artificial systems operating together, following some collective common behaviour, but not so rigid that unexpected or undesirable events do not occur. We also seek a domain where there are technical developments planned or under way, in order to frame our investigation in real domain issues, rather than hypothetical cases. Finally we seek a problem domain which will support investigation without drawing us into domain specific development (i.e. specialised interfaces, parsers, representation) or findings (where we lose application to the general problem issues). Against these criteria, we select the field of vehicles and traffic, in order to provide a domain which is rapidly evolving, contains numerous actor types, and where technological advances suggest we are heading into a disruptive period. The potential array of upcoming challenges posed by this domain align with those outlined as being of interest to this thesis, and so proves a suitable area for investigation. However, the aim of this work is not to provide specific solutions for the field of autonomous vehicle work, instead we are motivated by the more generalised questions, and seek to remain problem domain agnostic.

To express our goals in one sentence: this work considers the hypothetical introduction of some autonomous technology, into a scenario interacting with human users, in some complex environment. We pursue this goal in order to identify the difficulties that artificial intelligence systems may face, and for these situations propose approaches which may help resolve issues encountered. We now break this general problem area into constituent issues, allowing us to build up the facets of the problem case.

For an autonomous system to make decisions regarding action selection, there needs to be some element of ‘understanding’ regarding the environment in which it is situated. The term ‘understanding’ can be used with a variety of meanings, from attempting a human-like cognition process, to simple (reactive) procedural rule firing. We suggest that there should be an *appropriateness* between complexity and required solution, i.e. not to over-engineer a solution, but to adopt a suitable level of sophistication that is capable of meeting the requirements. This

forms part of the problem statement, as we need to identify, and prove the use of, an appropriate software approach to: codify the human knowledge of what should happen for given events, allow the software system to pursue goals given this knowledge base, and be able to provide a human observer with some explanation about its actions. However, in this work we limit the investigation into how humans may understand agent behaviour, considering the observer role only in the sense of debugging the system. For this case, the debugger is likely to be aware of the agent processes and terminology, and we seek to develop a system where observability of agent states is possible. We consider the larger issue around communication in general, and what to do when a situation occurs that the knowledge base does not cover?

Regarding communication, as well as being able to provide explanations for (and reasoning process behind) action selection, such systems need bi-directional communication with other (software and human) members across a broader range of topics, in order to plan around the actions of others. But, this raises the question of how much communication is required? To an extent, an exact specification of requirements could be considered domain specific (for example, air traffic control vs maritime shipping planning). Taking this example further, there is also consideration required around what ‘type’ of information is exchanged, for example an aircraft has to file a flight plan before take off to inform of its destination and route intent, but it also has to provide position data when travelling along this route. This touches upon a problem around communication, that to be able to enhance understanding of the current situation, information is required at both an appropriate rate, and appropriate level of complexity to represent adequately the desired knowledge.

Regarding events not covered by the knowledge base, the problem here lies in the fact that for a complex multi actor system, it would be difficult to encode every possible event and permutations of events into some knowledge store prior to runtime, especially with the introduction of human interaction leading to issues arising from cultural contexts, ambiguity in gesture and language, and so on. Furthermore, this is compounded by situations in which the autonomous system may need to coordinate actions with other actors (and vice versa), which raises the matter of how tasks related to coordination can be enforced on other entities.

## 1.1 Problem statements

We now summarise the various topics discussed so far, capture specific problem statements which we will use as reference points in the rest of this work, followed by specific objectives based on these problem statements in Section 1.2. These problem statements are revisited later in Section 3.2 where we expand further, but we provide the statements here to establish our initial views on the problem.

1. *How to build awareness and understanding of a situation:* An approach is required where autonomous entities can generate an appropriate level of understanding as to what is happening around them, in order to choose some plan of action to pursue their own goal(s).



2. *How can the reasoning process and information flow behind a decision be made observable:* With multiple, autonomous agents active in a realtime system, we envisage a problem where the human debugger needs to understand what has occurred in order to correct faulty behaviour. We see the need to provide observability of the AI ‘mindset’ in order to understand the causal chain of events, e.g. agent plan selection, why it has taken a certain course of action.
3. *Finding the right balance between quantity and appropriateness:* Here we focus on two areas, firstly *how much* to communicate, where too little communication can result in entities not knowing what is going on, but communicate too much and there is the risk of saturating communication infrastructure, or that others begin to ignore (or cannot process) the information flow and miss key pieces of information. There is also the question of *what* to communicate; and whether communicating more complex, rich information can circumvent the need to communicate a high quantity of ‘smaller’ facts. For example, if an entity has analysed a large amount of traffic data, and drawn the conclusion there is congestion ahead, is there benefit in just communicating that conclusion, or does the whole data set need to be sent.
4. *How to resolve problematic situations:* In this problem statement, we are concerned with situations where an agent encounters difficulty in achieving a task. We consider a group of cases especially relevant to agents working amongst a mixed human-artificial entity collective, where complexity arising from cultural contexts and ambiguity of human communication exacerbates the difficulty for an agent to know what to do. In other words, cases where there is no appropriate plan or action available to the agent for a given situation. This may be made worse if the artificial entity is unaware of its lack of knowledge, adding to the problems faced in this area.
5. *How to balance individual pursuits versus the greater social welfare:* Having selected autonomous agents which can pursue their own goals, the need may arise to influence the agent’s behaviour in order to prevent the sole pursuit of its own goals, in cases where these are to the detriment of the larger society. We see this as a need to contextualise the appropriateness of an action rather than provide agents with hard limits, for example it may be acceptable to break a rule (e.g. to drive through a red light) for a given context (e.g. to allow an emergency vehicle past).
6. *How to demonstrate the application of the principles we have identified, whilst retaining a general solution:* A side effect of wishing to investigate these problems and demonstrate any potential solution to this problem set, is that some implementation is required against a real world problem. Here a balance is required in identifying a suitable domain and appropriate scenarios, against the proposed framework becoming domain-centric.
7. *How distributed systems can support a variety of component types for a generalised set of applications:* The previous problem statements consider a society of mixed human and

artificial entities operating together in some environment, and in this problem statement we face the issue of how can we bring components together to support any required experimentation into this broad problem set. Here, we consider issues affecting distributed systems, as we wish to establish a framework where components can be introduced to provide specific capabilities (e.g. an intelligent agent framework, representation of a particular problem domain) without creating specific inter-component dependencies (e.g. replacing one component should not require rebuilding other components, or that one component always has to be present in order for others to function).

## 1.2 Objectives

The aim of this research is to address the problem statements introduced earlier in Section 1.1, and in doing so establish a generic framework where intelligent agents are able to pursue some goal(s), with appropriate communication and understanding, combined with some governance framework to both assist and manage their autonomy. Whilst these problem statements identify that we need to maintain a generalised solution, we ground the approach in the specific domain of intelligent vehicles, in order to assess how well the developed framework performs against realistic real world scenarios. These scenarios need to be constructed in such a way that they explore components of the problem statements, and so we identify the following objectives which they should address:

1. *Demonstrate appropriate intelligence:* The objective is to demonstrate a suitable intelligent layer operating in the problem domain. Whilst there are a wide variety of potential solutions, the objective is to adopt an approach where a higher level intelligence control layer provides direction to bring about the goals, in order to provide a level of abstraction away from the effort of low level entity control. This links with problem statements 1 and 2, and is investigated throughout all of the scenarios outlined in Chapter 5 as all of these draw on the intelligence layer (as described in Section 4.1.2).
2. *Demonstrate improved observability of reasoning and exchanged information:* To address problem statements 1 and 2, we wish to improve the overall observability of information held by components within the proposed framework, and of messages exchanged between these components. A key motivation is to expose the mindset of agents within the intelligence layer in a way which supports a human observer performing a system debugger role to understand the artificial systems' motivation and reasoning behind its action selection. The message exchange approach adopted to support this is detailed in Section 4.1.1, and a 3D viewing capability is presented in Section 4.1.7. The latter is used to display information extracted from the intelligence layer, which augments the 3D representation.
3. *Explore the effect of alternative communication strategies:* By adopting alternative communication approaches based on various levels of Situational Awareness (e.g. simple facts versus advanced comprehension of events), we seek to address problem statement 3

by exploring issues around the quantity and appropriateness of message exchange. The scenarios proposed in Sections 5.2, 5.3, and 5.4 form the main investigation, where intelligent driver performance is assessed whilst comparing the impact on infrastructure requirements.

4. *Provide guidance to agents for situations where they have no plan:* We envisage situations where socio-cognitive agents have no appropriate plan or ability to deal with a situation, due to the social and cultural complexity that can arise when interacting with humans. We propose a specific scenario in Section 5.5 where an intelligent vehicle requires assistance in determining what a signal from another vehicle means, and use this to address problem statement 4 through the exploration of this problematic situation.
5. *Demonstrate the curtailment of an intelligent agent's autonomy for the greater good:* Considering a collective of intelligent agents, there arises the possibility that agents may pursue selfish goals which need to be curbed, and also that coordination of the collective behaviour may produce a wider benefit than individually driven action selection. The scenarios of Section 5.4, 5.6 and 5.7 address this objective, presenting domain specific benefits such as improved fuel performance, as well as more generally applicable benefits through the demonstration of multiple institutions acting to resolve a complex situation, and are used to explore problem statement 5.
6. *Generate meaningful results in the domain of intelligent vehicles, based on a framework which can be used in other areas:* In this work, we investigate problems faced by artificial socio-cognitive systems, based in the domain of intelligent vehicles. We use this domain as a means to test our approach, rather than *define* it. Here the objective is to both demonstrate credible results being extracted from the problem domain using the proposed framework, whilst ensuring the solution and the domain do not become coupled, in order to provide findings in relation to problem statement 6. The proposed scenarios of Chapter 5 focus on the domain of intelligent vehicles, and have been presented to that community for discussion and feedback to address the credibility of the results, whilst in Chapter 4 the framework is designed to be reusable, with some initial work to integrate alternative end devices presented.
7. *Develop a distributed systems solution applicable to a general set of problems:* The objectives so far have already introduced the concept of different components that will be used in this work, an intelligence layer, a governance structure, and a 3D view capability. With this objective, we aim to demonstrate these working in a distributed fashion as separate components of the proposed framework, in order to address problem statement 7 directly, but also in relation to problem statements 2 and 3 where we wish to ensure communication between distributed components remains observable and that the supporting infrastructure is able to handle the volume of message exchange. Testing of distributed message exchange is presented in Section 4.2 and with tools to support the analysis of

exchanged messages in Section 4.1.6, and these are used throughout the scenarios put forward in Chapter 5.

## 1.3 Contributions

In addressing the objectives set out in Section 1.2 investigation of the problems set out in Section 1.1, this thesis makes the following contributions:

1. The application of Situational Awareness concepts to a distributed knowledge problem. A number of experiments are performed which explore aspects of situational awareness in relation to intelligent vehicles: in Section 5.2 we explore alternative message exchange rates of low level perceptions and the impact on convoy performance, in Section 5.3 we consider the alternative of exchanging higher level route information and its impact on the convoy, and finally in Section 5.3 we explore high level SA in the projection of the future state of traffic lights in allowing a vehicle to adapt its speed in order to arrive at those lights whilst they are green. We consider the contribution of these experiments to highlight how considering knowledge exchange in SA terms can improve performance in the vehicle domain, whilst reducing the communication burden.
2. Benefits of exchanging semantically annotated data. We demonstrate the dissemination of knowledge via semantically annotated communication where distributed components are able to consume messages of interest and reason using the additional data definitions within that message. For a human debugging the system, we consider the benefit as ‘Augmented Debugging’ where disparate data sources (e.g. agent mind-state data, network performance measures) are brought together to present a more complete representation of what is occurring across the framework. The semantic annotation is used by components to identify messages of interest from larger data streams based on that component’s own criteria (e.g. who published the message, what the message type is, when the message was created).
3. The application of institutional governance as a means of assisting intelligent agents. We show institutions providing guidance to vehicle agents in the scenarios of Section 5.4, 5.5, and 5.6, where they resolve ambiguity, tackle problems of coordination and promote societal benefits over individual again. These are demonstrated in the vehicle domain by resolving a flash of headlights (as indicating another wish to overtake), as a means of managing traffic flow through improved traffic light awareness, and the finding that this leads to reduced emissions and fuel consumption. We ensure general application remains through the construction approached outlined in Chapter 6.
4. Multiple institutions interacting within a realtime environment to resolve a complex situation. We demonstrate multiple governance structures working together in the resolution of a post-accident scenario which is presented in Section 5.7. Multiple bodies (insurers, emergency services and motorway management) work together to issue appropriate

guidance to the intelligent entities within this scenario, to coordinate management of the accident (through merge lane behaviours) as well as post-accident logistics (dispatching emergency units and insurance claims).

5. The demonstration of a generic framework capability. We show the developed framework used in the chosen problem domain of interest in Chapter 6, whilst retaining application to other areas through the adoption of generic components such as governance structures, 3D representation, an intelligence layer and analysis tools, configured for a specific domain through (external) data files instead of embedded code, based on an open approach to communication of semantically annotated messages.

## 1.4 Thesis structure

The remainder of this thesis is comprised of the following chapters:

### **Chapter 2. Related research**

This chapter considers research which has influenced or underpins the direction and arguments of the work presented in this thesis. The topics touched upon by this thesis include fields from human psychology, computer science, and traffic domain specifics. As such, there has been some trade off between depth and breadth; where a design decision has been taken for a specific approach (e.g. intelligent agents) related research is presented around this area rather than considering all possible design solutions.

### **Chapter 3. Analysis of the problem**

In this chapter, the various areas of the problem domain are expanded upon further with consideration of the general problem areas, followed by a number of specific problem statements are developed. The requirements of the problem domain are then considered, followed by a generic set of requirements which a suitable architecture would need to provide.

### **Chapter 4. Developing an architecture and software solution**

This chapter presents the work undertaken to develop a framework capable of addressing the problem statements set out in this work, and presents details of messaging framework, the intelligence layer and key agent capabilities, the investigation and implementation of entity representation, the institutional framework used, the 3D representation capability and general framework utilities.

### **Chapter 5. Implementation of scenarios**

In this chapter details are provided of the scenarios which have been developed to investigate the problem statements set out in this work, exploring issues of communication, knowledge and understanding, resolving challenging situations through providing the intelligence layer with

the additional support of the institution framework, and then using this institution approach to manage the greater collective benefit versus individual pursuits.

## **Chapter 6. Analysis of results**

In this chapter, results of preliminary work are presented where relevance was found to the problem statements addressed in this work, before presenting details of the metrics used to assess performance of generic communication measures (e.g. quantity, delay) as well as domain specific vehicle and traffic measures. This is followed by the presentation of the results generated by the scenarios, which are then considered in the generalised terms of the problem areas we are focussing on.

## **Chapter 7. Further work**

This chapter presents a number of themes of further work, discussing domain specific areas of improvement, followed by thoughts of where adding additional features into the domain such as human in the loop drivers, and alternative (real) vehicle types could generate additional findings. Finally, broader questions are considered of factors affecting trust in socio-cognitive systems, how to manage varying communication approaches in realtime, and the violation and temporal elements of institution guidance.

## **Chapter 8. Conclusions**

Finally, conclusions drawn from this work are presented, where the problem statements are revisited and the findings against these, as well as corresponding objectives, are discussed.

# **1.5 Related Publications**

The following publications have been produced during the development of this thesis:

[Lee et al. \[2012\]](#) - JeeHang Lee, Vincent Baines, and Julian Padget. *Decoupling cognitive agents and virtual environments*. In Cognitive Agents for Virtual Environments - First International Workshop, CAVE 2012, Held at AAMAS 2012, Valencia, Spain, June 4, 2012, Revised Selected Papers, volume 7764 of Lecture Notes in Computer Science, 2013, Springer International Publishing. Pages 1736. ISBN 978-3-642-36443-3.

[Baines and Padget \[2015\]](#) - Vincent Baines and Julian Padget. *A situational awareness approach to intelligent vehicle agents*. In Michael Behrisch and Melanie Weber, editors, Modeling Mobility with Open Data, Lecture Notes in Mobility, pages 77103. Springer International Publishing, 2015. ISBN 978-3-319-15023-9.

## Chapter 2

# Related research

In Section 1.1 the problems this work seeks to address were put forward, along with objectives of the work (Section 1.2), providing direction and scope for research areas to be considered. In this chapter, we consider related research areas, and draw on approaches and examples where existing work has tackled or is related to the problem statements of Section 1.1. As focus is on artificial socio-cognitive systems, which we consider in the domain of intelligent vehicles, there are a large number of potential topic areas, and so we adopt an approach of drawing on pertinent related pieces of work, focussing on: i) human factors, ii) computational issues, and iii) the problem domain..

One of the key areas we identified in Chapter 1 is that of understanding, and in this chapter the topic is expanded upon in the area of situational awareness, with examples of the importance of such awareness provided. Following this, the topic is broadened to discuss *shared* situational awareness, considering how individuals can exchange information to improve both their own, and others situational awareness, in order to improve the overall group understanding of what is taking place.

Following these human factors focussed topic areas, we discuss work in the area of intelligent systems from a computer science view point, considering issues around the autonomous system aspect of the problem domain, as well as the ability of such systems to fit with the situational awareness and communication needs. We then extend this to consider how the autonomy of intelligent systems can be managed, given the problem statements of Section 1.1 regarding the resolution of problematic situations and managing individual pursuits versus the greater collective welfare, where we review the use of institutions to provide such capability.

Finally, we consider work from the proposed problem domain of autonomous vehicles. Although stated in Section 1.1 that we wish to retain abstraction from a specific problem domain, significant effort was required to place scenarios meaningfully and accurately in the context of vehicles and traffic simulations. This is to avoid both the ‘garbage in garbage out’ syndrome of misleading results, and also to demonstrate meaningful benefits and application of this work to the intelligent transportation systems community.

Relating to these topics, we firstly declare our use of two terms, which will occur frequently

in this thesis: *knowledge*, and *information*. The Oxford dictionary states knowledge to be:

*Facts, information, and skills acquired through experience or education; the theoretical or practical understanding of a subject*

By comparison, the Oxford dictionary states information to be:

*Facts provided or learned about something or someone*

We draw from this, that it could be defensible to use the two terms to mean the same thing, and that to make a clear distinction between them is somewhat challenging. For example, a raw sensor feed might be considered information rather than knowledge, but if this goes through some data fusion, is it still ‘information’ or has the additional richness transformed this into ‘knowledge’? Knowledge perhaps implies some procedural awareness compared to information, but if that procedure can be broken down to a set of facts, then by definition it is still information.

Having set these terms out, we now consider the first grouping of related research, from the field of human factors.

## 2.1 Human factors

Within the topic of autonomous systems, functioning in some intelligent fashion, interacting with both software devices and humans, there is a broad topic of human-related research areas to consider. In this section, we group related research into two main areas: situational awareness, and *shared* situational awareness. In the case of the former, we consider this a useful category encompassing aspects of the problem such as how understanding of the environment and events can be interpreted and for the latter we see this as a progression towards how interaction between agents in the environment can influence both the groups’ and the constituent individuals’ understanding i.e. a shared situational awareness.

### 2.1.1 Situational Awareness

Situational Awareness (SA) is a term used to refer to an entity’s understanding of what is happening an environment, and as a first approximation, the definition put forward by Adam [1993] that it is “*Knowing what’s going on so you can figure out what to do*” provides a succinct summary. However, to expand upon this and add some features which relate back to areas of the problem domain put forward in Section 1.1, we draw on the definition put forward by Endsley [1995], defining Situational Awareness as:

*“the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.”*



From this definition, we first consider “perception of elements” as the ‘low level’ registration of events and aspects of the environment in which the entity is situated. These are considered as simple facts, arriving to an entity via some sensory feed but with little or no additional meaning (e.g. a light is on, I’m at position xyz, today is Friday). From this ‘low level’, we move up a level in an information hierarchy to the “comprehension of their meaning” component of SA, where some additional information exists to augment the low level perception. Taking the position perception example, this could be quite specific such as ‘previously I was at x1,y1,z1 so I have moved 10 metres’ or more general ‘position xyz is not where I want to be’. Finally, we move to ‘high level’ SA, “projection of their status in the near future”, extending beyond just comprehension of perceptions to what that means in some future time. Using the previous position example again, this could be ‘Having moved 10 metres, at this speed I will collide with an object in 10 seconds’. In [Endsley and Jones \[1997\]](#), the importance of mental models (to represent possible situations) is highlighted as the means to transition between low level (1) SA to higher level (2 and 3) SA, using an example of recognising that a group of aircraft (from level 1 SA) are flying a particular formation (level 2 SA) by pattern recognition against prototype mental models.

This notion of ‘low level’ and ‘high level’ SA appears frequently in the rest of this thesis, as a mean to indicate the level of understanding contained in information or knowledge held by an entity, but also being exchanged with other entities. This provides a shorthand means of referring back to these SA concepts when discussing entities exchanging high or low level information, to indicate the complexity and richness of the exchanged messages.

To improve understanding of the environment, we consider how these SA layers can be built up from generated perceptions, using examples based on the vehicle domain. At the ‘low level’ perception layer, a comparison between human sensory systems and physical sensors can be considered, with imperfections and inaccuracies in the data received from them (eye defects, hearing problems) of which the individual may or may not be aware of. When these sensory inputs are aggregated to derive a comprehension of what is happening in the environment, another tier of variability is added with any errors compounded during the aggregation. Expanding this to several entities, two individuals studying the same environment could hold differing SA, leading to issues with shared situational awareness (discussed more fully in [Section 2.1.2](#)) as to how such variance should be handled. For example, in the context of the problem domain, this bears relevance as different (agent) drivers may hold a differing view on their projection of future events (and so take different, perhaps conflicting, actions), and highlights the need for improved handling of such cases, which we intend to address through the problem statements set out in [Section 1.1](#) regarding observability of a decision making process, and communication approaches. Agents need to be able to explain their decision making process, both to allow human observers to understand why something happened, but also to enable a coordination and resolution role to be possible when conflict occurs (e.g. a hypothetical case where Agent1 needs to turn left because of PerceptX, Agent 2 needs to turn right because of PerceptY but both are blocking each others turns).

As well as a convenient structure to analyse the understanding aspects of the problem, SA can be used as a means to consider ‘what went wrong’ in our autonomous agents behaviour, and has been used in such real life analysis. Here we draw on investigations into aircraft related incidents, firstly with work by Hourizi [1999] in relating Endsley’s three components of SA to failures in understanding the current state of an aircraft, given as:

1. failure to perceive important elements in the environment;
2. failure to comprehend the elements that have been perceived;
3. failure to predict the future status of those components.

As SA can be likened to human understanding of the environment, and it is this which informs human decision making, then it follows that an incorrect or lack of SA can be the cause, if incorrect actions or decisions are taken. There is no self-awareness as to SA being accurate, complete, inaccurate, or incomplete. Whilst we introduce this notion from a human perspective, it also needs consideration in the artificial systems domain, where with a number of layers generating an environmental view for a system, if one of these sensed values or inferred meanings is incorrect, it raises the question of how the artificial system should detect this. Whilst critical systems can employ a voting mechanism to detect and ignore faulty sensors (i.e. low level SA perceptions), it is challenging for a similar reflection on the higher level SA (e.g. comprehension, projections) of sensed data, and thus the appropriateness of action selection based on the perceived state of the environment. This issues links back to several of the problem statements of Section 1.1: we wish to communicate more meaningful data to allow systems to reason about what they have been told, we wish to make their reasoning process visible so others can review it, and we wish to develop some governance structure that can intervene if an entity is taking inappropriate actions.

Following air accidents, there is an in-depth post-accident analysis process, which provides a rich, detailed source of information and analysis as to what went wrong. There are a number of aircraft incidents which have been attributed both to the pilots’ incorrect understanding of what was happening (e.g. d’Enquêtes et d’Analyses pour la sécurité de l’aviation civile [2012] report into Air France flight AF447, pilots suffering a “total loss of cognitive control of the situation”), and their contrasting belief that what the aircraft instruments are telling them cannot be correct (i.e. that *their* SA is correct) leading to disastrous action selection. Specific to the vehicle problem domain, it is not difficult to relate many car crashes to similar conditions; any accident report with a causal statements such as ‘I thought he/she was going to set off’ signifies that the accident could be linked with a driver having an incorrect belief of how events are going to unfold (i.e. projection element of SA) and selecting an action based on this (i.e. continue at same speed and do not decelerate).

It would seem that humans are susceptible to reaching a conclusion that their SA is sufficient and correct, and despite contradictory evidence becoming apparent, deciding that the validity or source of this new data may not be reliable. More detailed consideration of this is presented later in discussions around shared situational awareness in Section 2.1.2.

Some explanation of incorrect SA can be found in research which suggests that humans do not like reasoning under uncertainty (in general, in the field of neuro-economics, and more specifically in [Huettel et al. \[2005\]](#)), coupled with having a limited working memory capacity (i.e. ability to process and hold large data sets). For this reason, they seek to resolve the issue by reaching a (perhaps hasty) decision as to what is happening in the environment. In contrast, computer systems do not suffer from such limitations, but are differently challenged, being limited by the inferential capabilities with which they are designed, but well-suited for other activities (e.g. large data mining and processing activities). This creates a potentially strong application for SA in AI, being well suited to continually re-assessing its conclusions based not only on newly-received data, but a wider permutation set of how future events may occur.

Such an AI-SA implementation could be taken further, for example identifying where additional information would prove or disprove the current assertions and hypotheses and thus trigger the system to seek such data (e.g. reorient a sensor), or where reflection upon currently held SA may suggest incompatible beliefs (e.g. speed is zero but position has just changed). If contradictory data is received, then reasons for this could also be explored e.g. something may be happening which the system does not understand, there may be some anomaly with the data itself, or there may be actions taking place in the environment which are seeking to deceive the observer. This theme also relates to the issue of reliability and trustworthiness of information, and whilst we seek to develop a system which is capable of addressing such concerns, we do not explicitly include any variability regarding such factors in this thesis, but we highlight this as interest for future work. We revisit the potential benefits of such a reflection capability when considering intelligent architectures later in Section [2.2.1](#), based on the potential benefits this would have, and the associated need to select an architecture which is able to reason about its understanding in such fashion.

If we consider SA to be improved through attentive engagement with the environment, then it seems logical the inverse holds true, that by being inattentive and missing potential perceptions, SA is degraded. Considering this in a vehicle domain, for human drivers there are plenty of examples of driver distraction (e.g. using a phone while driving) causing accidents. In [Kern et al. \[2008\]](#) a driving simulator (based on the 3D software presented later in Section [4.1.7](#)) is constructed in order to measure driver distraction over a given virtual course. The work in [Kern et al. \[2008\]](#) focusses on the role of user interfaces in driver performance, though their metric of performance is how well the route is followed rather than developing any metric of driver attention of SA. The implementation in [Kern et al. \[2008\]](#) also lacks environmental features and background traffic, and so it becomes difficult to judge how much this simulation reflects a real life situation. This serves as a reminder, that for the problem statement of Section [1.1](#) regarding understanding the environment, there is risk of oversimplifying, and there being insufficient complexity in the environment to really test the intelligence and reasoning of artificial entities (a topic relevant to the problem domain representation discussions of Section [4.1.3](#)).

In conclusion, we draw on Situational Awareness as a framework with which to consider the levels of information exchange taking place between intelligent artificial systems, which we use

to frame our response to the problem statements of Section 1.1 concerned with how to build awareness and how to find the balance between quantity and appropriateness of communication. We have provided examples of how causes of accidents can be considered as a lack of, or incorrect SA, and this creates motivation to select an intelligence framework which can reason based on information at various (SA) levels. Finally, we consider that the environment in which this intelligent system is assessed needs to provide a suitable level of complexity in the problem domain of intelligent vehicles.

### 2.1.2 Knowledge sharing

Having introduced SA at an individual level, we now expand the discussion to consider how this awareness can be shared and developed amongst a collective, through appropriate communication and knowledge sharing, as a ‘shared situational awareness’. For this, we discuss both how understanding can be generated, and how it can be communicated. We consider this as an information flow: a ‘fact’ or percept first needs to be generated, before it can be shared. Once this has been achieved, there is the question of how it should be communicated. Finally, there is a feedback loop; having been told about something via a communication, a decision is required on how to update the view of the environment for each individual in the collective receiving this information.

#### Understanding

The broad topic of ‘understanding’ (and indeed the meaning itself) is vast, and so focus is placed on topics considered to be related to intelligent systems, and also to the problem domain of transportation. In discussing understanding, we draw on a definition from the Oxford dictionary:

*An individual’s perception or judgement of a situation*

This links to the earlier discussions regarding SA; that there is a connection between perceiving elements of a given situation or environment, and faulty or missing perceptions that are likely to affect the generated understanding. Here we draw on the problem statements introduced in Section 1.1 regarding understanding of the environment, and highlight that this understanding needs to be collective rather than just individual, otherwise it will be difficult for entities to interact in an environment where they have different SA and so expect different events to happen. There is also an implicit question as to whether an artificial system can generate acceptable understanding in a human-dominated environment.

Initially, we consider both of these issues in combination, when looking at how and why differing SA may be generated by two observers looking at the same situation. As well as being influenced by cognitive processes, there is also some sensor system involved, and as SA becomes built up from this ‘feed’, there is a weakness in suffering short-comings in those sensors, as well as unexpected nuances of how data is processed.

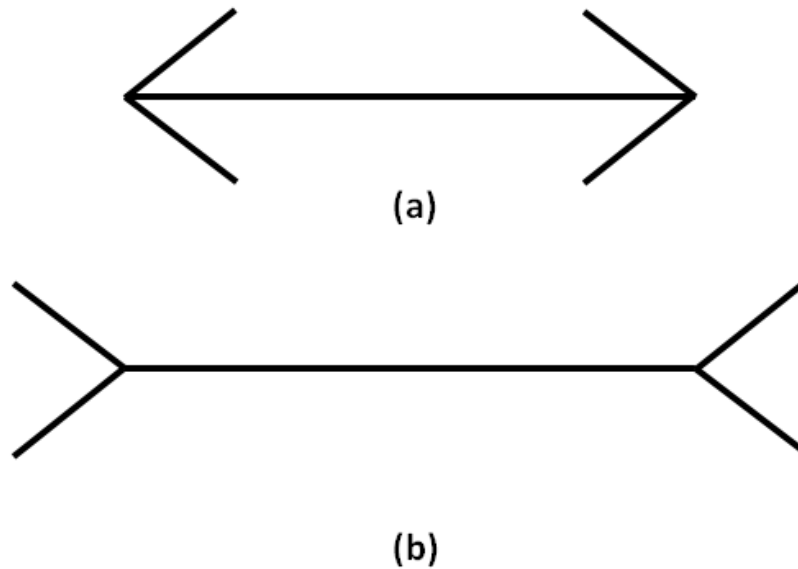


Figure 2-1: Müller-Lyer Illusion

Our understanding of the environment would seem to be heavily influenced by expectations, demonstrated by the Müller-Lyer [1889] Illusion shown in Figure 2-1. While both lines are the same length, the addition of inward or outward facing arrows gives the impression that (a) is longer than (b). One explanation for this affect is that we are conditioned to expect additional geometry to indicate our proximity to the line and then infer distance to the object. The arrows in (b) suggest that the line is closer to us than the arrows in (a), and hence the illusion that it is longer.

To connect some intelligent system to real world sensors, operating in an environment populated by human entities, is a challenging problem. Specific to the work undertaken here (phrased in terms of vehicles being driven by humans), the scenario introduces a need for more than low level SA (e.g. perceiving that an indicator light is on) to higher level SA in projecting the future impact of this (in this case, that the vehicle will change direction). As suggested, humans can indicate this to other humans through indicator devices on the vehicle (e.g. indicate left before turning left), allowing other drivers to plan around their indicated future state. However, humans are also fallible, and at times leave their indicators on – which humans rapidly perceive as not signalling what the indicator normally signals – or forget to use them in the first place. Their communication may also be unclear, for example a flash of headlights in some situations means ‘you can pull out’ but in other situations means ‘let me past’. Whilst there is typically a ‘rule book’ for such indications (in the UK, ‘The Highway Code’) this is not always adhered to (or enforced), and cannot cover all eventualities. Such

problems are touched upon later in Section 2.2.2, where the use of institutions is explored as a ‘reference source’ for problematic situations.

Having considered the understanding of just one piece of information or percept (e.g. a flash of headlights), we move on to consider the effect on SA of receiving a vast, simultaneous data feed, i.e. operating in the real world where, across multiple sensors, there are multiple percepts occurring in real time. Some aspects of this are discussed in the following communication section, however one aspect related to ‘understanding’ is the ability to create ‘higher level’ (in the SA sense) knowledge from a large volume of ‘low level’ data pieces.

Such activity could be viewed as a data fusion problem, meaning that there are a wide number of potential solutions which could be adopted. We consider stream reasoning (Anicic et al. [2010], Barbieri et al. [2010], Gebser et al. [2012]) as one interesting approach, where the input is a high volume data stream, which is distilled into ‘high level’ information to contribute additional knowledge to SA. For the domain of vehicle traffic, this could be a data stream containing all vehicle positions and speeds on a given route, and determining whether congestion is going to be encountered (effectively transitioning from a huge quantity of low level perceptions, to a high level projection of future state). Of particular interest is the work by Ranathunga et al. [2012], where low level information (such as coordinates, velocities) is extracted from a virtual environment (Second Life) and fed to a Multi Agent System (MAS) with appropriate triggering of plans based on higher level derived understanding. From this, we draw the positive conclusion that it is possible to manipulate a high volume, low level perception feed, into a form suitable for agents to act upon in a more high level fashion. We adopt a simpler, less formalised approach (which we present in Section 4.1.5) than that put forward in Ranathunga et al. [2012], based on the premise of supplying entities with higher level information based on a geographical area of interest centred at that entities current location. Although lacking the sophistication and general application of approaches such as stream reasoning, the motivation in the problem statements of Section 1.1 to develop a generalised distributed solution means we develop an approach where it would be straightforward to replace the developed solution of Section 4.1.5 with an alternative approach.

Finally, to add some intelligent vehicle specific developments, of interest is a recent statement by Urmson [2014], one of the developers involved in the Google Car project, that their vehicle can now “detect hundreds of distinct objects simultaneously” including “a cyclist making gestures that indicate a possible turn”. In the context of understanding and SA, this proves an interesting development, and in exploring the problem areas outlined in Section 1.1, allows us some assumptions: systems are now capable of processing hundreds of objects per second, and that complex events (a cyclist presumably moving their arm in a certain pattern, detected over a number of camera frames) can be reported at a much higher level (e.g. cyclist1 turning right). This allows us the luxury of side stepping the technical challenges in processing such data in the real world, and instead reporting it via simulation for our experimental purposes. It also adds weight to our selection of intelligent vehicles as a domain to address the problem questions: the technology *is there*, now what could a framework around it look like?

## Communication

Following on from the discussion of SA and understanding, is the closely related topic of communication. With the topic of SA discussed in Section 2.1.1, we considered varying degrees of information ‘richness’ as levels, and in the discussions about understanding discussed in Section 2.1.2, we considered issues relating to perceptions and touched upon the topic of data quantity. In respect of this, the question arises of how much information needs to be passed between distributed agents, and at what level (in the SA sense) this information needs to be. Finally, there is a question of how this information should be communicated, introducing real world issues such as network capabilities, but also how the information should be encoded both at a serialization level and a ontology level (i.e. do we just assume the recipient knows we are sending information about speeds with units of miles per hour). Essentially, we need to address the questions of: How (do we exchange data)?, What (do we send)?, and When (at what frequency)?

Work by [Chen \[2003\]](#) proposes only to communicate information which is needed and beneficial to other agents, and considers communication requirements involving cost and lack of bandwidth in long distance communication for a planet exploration scenario. The reasoning about this information exchange seems to be handled at an individual agent level, where knowledge of the agent teams’ plans is used to identify agents which require information to be passed on, thus reducing the amount of communicated perceptions. The concept of collective intelligence is also discussed, observing that it is straightforward for agents to exchange quantitative data about their situation (e.g. position, goals, status, etc.), and that superfluous data can be ignored.

However, whilst reducing communication where possible is a sensible goal, there is difficulty in identifying what counts as superfluous, and what could be of use to another agent. Furthermore, there is the issue of where this decision can be made, as the potential recipient may wish to decide this, rather than the potential sender. Such communication requirements also vary depending on the nature of the participant, humans may find high levels of communication tedious, as well as potentially distracting (and therefore a negative affect on their SA). An autonomous system may be better suited to such communication, and able to derive a benefit in SA compared to the negative impact on human SA. For the problem domain of transportation, a relevant example is reported in the press ([Moskvitch \[2011\]](#)) demonstrating that vehicle communication could lead to significant benefits in reducing motorway pile-ups. Similarly here, there is a question of to whom should this information be communicated, and where does the decision lie as to who would be interested in such updates. In [Harding et al. \[2014\]](#) two significant results regarding the potential safety benefits of such vehicle to vehicle communication are shown, firstly in a pilot where vehicles had required communication devices fitted and were able to exchange messages, and secondly that it outlines a number of manoeuvre types where they target vehicle to vehicle communication as a means to improve road safety, and we reproduce these groupings here:

1. Forward Collision Warning: Warn if a rear-end collision with the vehicle ahead is pre-

dicted.

2. Electronic Emergency Brake Light: Warn if a vehicle further ahead is performing a hard brake.
3. Do Not Pass Warning: Warn if unsafe to attempt an overtake.
4. Left Turn Assist: Warn if turning across a lane would result in a collision with oncoming vehicle.
5. Intersection Movement Assist: Warn if unsafe to enter an intersection.
6. Blind Spot Warning and Lane Change Warning: Warn if vehicle detected in blind spot when attempting to change lane.

We highlight these cases as although they are based in the specific domain of vehicles, the type of information being exchanged has a more generalised implication: These cases could be considered as ‘mending’ an entities incorrect SA. If an entity had a high level of (correct) SA, then they would be aware that an action selection is inappropriate (e.g. being aware there is a vehicle in their blindspot). From this, we draw two points, that there is a benefit of communication where an event may be outside of an entities perception range (i.e. a vehicle braking far ahead), and when some perception, comprehension, or projection has not occurred and an entity is ‘missing’ some realisation about their environment (e.g. the vehicle which was approaching from behind is now in the blindspot).

With this motivation to improve communication, we first consider the approach of a data-push, with information sent out to any listeners who have registered an interest in such updates. This places the responsibility (and control) on to the individual entity: if they are capable (in processing, bandwidth, etc) of handling high volumes of perhaps ‘low SA’ data, then they could use that feed. If they are only interested in making use of ‘higher level’ SA type information, that would be an alternative data source. However, whilst this may offer some choice over volume and type of information, it does not necessarily mean that the received data will be immediately understandable to the recipient. To address this part of the problem, we consider developments of “The Semantic Web” ([Berners-Lee et al. \[2001\]](#)) as an approach to semantically annotate the information being sent. This seems well aligned with the SA approach we have put forward; the sender of information presumably has the knowledge of what this information is about, and therefore by adding descriptions about that data (unit definitions, the sensor which perceived the data, time it was created at, etc) creates a much richer piece of information to share. This gives the recipients significantly extra empowerment: not only can they decide which channels they wish to subscribe to, but can discard individual messages which are of no interest based on the semantic annotation. Furthermore, it facilitates reasoning about received information in order (ideally) to transition the data from the low level perception layer, to higher levels of SA. It may also assist in the exchange of SA components at a level higher than perceptions, where a recipient may not know what is going to be sent beforehand, but be able to make sense of the received information through the semantic description. In [Section 2.2.1](#) we discuss the agent communication languages of KQML and FIPA, and we highlight the view here, that whilst we consider semantically annotated messages benefiting *inter*-component exchange, there is likely to be justification in adopting alternative *intra*-component message exchange



approaches. Where existing standards of communication have been developed for a specific task and domain, the effort and potential loss of functionality in adopting semantic messages may make migration to an alternative communication format unwise.

Finally, in terms of the high level problem definition, it would seem a good approach for any future technology to offer the capability of better integration with devices around it, as the “Internet of Things” coined by Ashton [2009] suggests we are heading towards a much richer (information and sharing), connected world. In consequence, a review of related technologies which would support such communication follows in Section 2.2.3.

### Shared Situational Awareness

We now discuss the topic of Shared Situational Awareness (SSA) as an extension to the SA concept, which combines the elements of communication and knowledge exchange within a collective, rather than just focussing on individual awareness. It also considers the coherence of that understanding amongst the group, that is, whether for a given situation all members have the same understanding of what is going on.

Early work on this concept was focussed heavily on military aspects, where the importance of individuals within a team sharing the same understanding of what is happening is crucial (e.g. Court [2006]). Shifting away from a purely military focus, Harrald and Jefferson [2007] consider SSA in the area of emergency management and response, and highlight problems affecting the generation of ‘good’ team SSA. Issues affecting SSA are put forward in Harrald and Jefferson [2007] including the semantic meaning of the data, the quality, selection and integration of the data and consistency of both perceptions and conclusions drawn from the data, and we draw on this in our investigation into a collective of socio-cognitive artificial systems, considering the problem statement of Section 1.1 of how to build awareness and understanding of a situation. By adopting a communication approach where data is semantically annotated, we seek to address some of the problems put forward in Harrald and Jefferson [2007] by improving the understanding of *what* is being communicated between the intelligent system members. However, this may introduce the risk of multiple points of failure (i.e. ‘mass group-think’ if all autonomous entities are consuming the same data and reasoning about it in the same way), relating to the problem discussed in Section 2.1.1 of self-awareness of accuracy and completeness of the self’s SA.

In summary, we consider the area of SSA as being challenging, in part due to the lack of a pre-agreed concise semantic message exchange format between humans. Whilst not in direct focus of the problem statements of Section 1.1, socio-cognitive systems would be likely to benefit by adopting the approach of exchanging more precise, semantically annotated messages, to improve their overall performance as a collective through an improved shared understanding of the environment.

## 2.2 Computational representation

Having established what we mean by Situational Awareness and how we intend to use that concept with regards to communication and understanding, we now move to consideration of the computational representation aspects of the problem space. In Section 2.2.1 we consider intelligent systems which are able to handle, exchange, and act upon information at a rich level, remaining abstract from the domain of transportation systems, whilst wishing to demonstrate effectiveness against scenarios in that area in order to demonstrate the validity of our approach.

Following this, in Section 2.2.2 we revisit the problem question of how such intelligent systems can be managed, and review related work which has considered the issues of organisation and governance, where the topic of institutions is introduced with examples of these being demonstrated in intelligent systems.

Finally, in Section 2.2.3 we attempt to pull these considerations of intelligent systems and their management, together with the earlier discussions of communication and understanding, to review what approaches may fit the requirement of information exchange for the approach of this research.

### 2.2.1 Intelligent Systems

In this section, we revisit one of the problem statements from Section 1.1 regarding building awareness and understanding, and consider what computational approaches might support such a requirement. Previously in Section 2.1 thought was given to using SA as a means of viewing knowledge of a situation at varying levels, and how communication can enable both the self and collective SA to be improved. Having set out in the introduction to this thesis our belief that autonomous systems will increasingly be used, we now review related technologies and solutions related to this area, focussing firstly on intelligent architectures and then intelligent agents.

Such a review is challenging as the options are vast, and the fundamental question of what constitutes ‘intelligence’ in itself is subjective, creating ambiguity as to what should be considered in the scope of this review. For this reason, we explicitly state the objectives of this section of the review, to: i) draw on existing assessments of intelligent architectures, rather than repeat existing work, whilst ii) identifying where these architectures may fit with the problem statements introduced in Section 1.1, before finally iii) drawing on work where intelligent agents have been used in the domain of intelligent vehicles.

#### Intelligent architectures

In Long et al. [2007], a review of intelligent software for autonomous vehicles is conducted, presenting a set of commonly used software systems, which we use as a start point for potential architectures. Amongst others, Long et al. [2007] discusses expert systems (JESS), fuzzy logic, subsumption architecture, hybrid deliberative and reactive architectures (AuRA), through to the cognitive architectures Soar (Laird et al. [1987]) and ACT-R (Anderson [2007]) (and we will

come back to address omitted architectures). However, no analysis of strengths or weaknesses of these approaches is discussed, instead they are grouped into a table to be compared in one location with details such as language, underlying technology, whether they are reactive or deliberative. It is acknowledged in Long et al. [2007] that this list is not exhaustive, and so we need to consider additional architectures, but before doing so we reflect on technology groupings provided in the conclusions of Long et al. [2007] : neural network, genetic algorithm, fuzzy logic, symbolic AI and learning. Given the problem statement introduced in Section 1.1 regarding the observability of process flow and reasoning (to a human debugger), there may be motivation to avoid the less (immediately) observable of these technologies, where the ability to trace back through a decision making process in human-like terms may be difficult in neural network, genetic algorithm and fuzzy logic approaches.

We draw on the problem statements introduced in Section 1.1 as a means to refine and focus our consideration of suitable intelligent architectures. The problem of building understanding of a situation pushes us towards a more sophisticated architecture, which when combined with the problem of making the reasoning process observable suggests an architecture which reasons in human-like terms. The problem statements regarding developing a generalised, distributed system also assist in defining potential candidates, as we need an extendible, open-source solution capable of performing in different domains. Considering the cognitive architectures discussed earlier in Long et al. [2007] as suitable candidates for this criteria, we first review their use in the domain of representing intelligent entities. Soar and ACT-R are both heavy-weight cognitive architectures, established for over 30 years, proving themselves in various problem cases such as controlling virtual entities in simulations (e.g. ACT-R with Unreal Tournament shown in Best et al. [2002], Soar aircraft shown in TacAir-Soar, Jones et al. [1998]). However, whilst these architectures are undoubtedly strong in representing cognitive processes, from the problem statements of Section 1.1 we have additional considerations concerning functioning with external governance structures and the interaction of multiple entities in a distributed environment.

In Norling [2012] the suitability of the Belief-Desire-Intention (BDI) framework is assessed for suitability in modelling human behaviour, which highlights the folk-psychology roots of the framework as a strength in supporting knowledge representation. Although we do not seek to directly model human behaviour in this work, from the problem statement of Section 1.1 regarding observability of the reasoning process and information flow, we need to be able to represent an artificial systems' reasoning in human-like terms that can be understood by the human debugger. The BDI model was put forward by Bratman et al. [1988], with formalization put forward by Rao and Georgeff [1991], and has lead to a variety of software implementations. It is an approach applicable for programming intelligent agents, handling the process described earlier of sensor feed back, processing, and action selection. Agents contain a belief base, usually updated via the sensor feed, and are equipped with a 'plan library' in order to achieve some goal, though with the drawback that if there is no plan available for a given situation, the agent is unable to react, which relates to the problem statement put forward earlier in Section 1.1 regarding handling problematic situations, as this includes an agent not knowing

what to do. This belief base can be reasoned about internally by the agent, but with regards to the SA discussions earlier, it is of interest to generate behaviour based on the receipt of higher level information. In [Hindriks et al. \[2011\]](#), the need to avoid overloading BDI agents with perceptions (which we have classified as low-level SA) is discussed, with the need to delegate through appropriate layers, i.e. that the BDI layer should perform higher level reasoning rather than suffering “cognitive overload” by handling high volumes of percepts. This approach is fundamental to the system design ethos we adopt, where framework components should be provided with data at an appropriate SA level, i.e. the intelligence layer should reason at a higher level than data fusion components described in Section 2.1.2.

[Evertsz et al. \[2008\]](#) demonstrates the use of a BDI agent in controlling simulated tanks in a virtual environment, which has relevance to the problem domain of intelligent vehicles considered in this research. In [Evertsz et al. \[2008\]](#) ‘CoJACK’ is used, a development of the JACK commercial software offering (which includes a BDI model implementation) which has been extended to introduce additional cognitive factors. Although the aim here was to add more ‘human variability’ to the scenario (in order to add richness to otherwise prescriptive JACK agent responses), it highlights that such approaches are feasible and have proved fruitful in the past. In [Howden et al. \[2001\]](#) a more detailed review of the JACK architecture is presented, which puts JACK forward as an effort to bring the Java flexibility to BDI agent behaviour, though diverging from the more AI specific programming approaches. The proposed “SimpleTeam” model introduces the capability of team-based reasoning, although this appears to be more of a mechanism to specify team behaviours and organisation than a distributed mechanism to improve shared situational awareness.

In [Shendarkar et al. \[2008\]](#) the use of BDI agents is presented in an emergency response and crowd simulation scenario. BDI agents are used to represent individuals, specifically how they react to danger threats and choose an exit path. The authors draw on [Norling \[2004\]](#) to highlight the benefit of folk-psychology in creating an easy mapping of BDI to human terms of reference, and we consider this in relation to the problem statement in Section 1.1, that such terminology may assist the human debugger in understanding what a BDI agent is ‘thinking’. Unfortunately, details are sparse on the specifics of the implementation, or what (if any) existing BDI model it is based upon, though their extension proposal of adding an emotion module within a situated BDI agent is interesting, and may serve to improve the representation of human-like behaviour. Their conclusion that this extended BDI model could be used in driverless cars is also relevant to the problem domain we are interested in, although no specific implementation plans are proposed in their work.

Following the discussion of intelligent architectures and the Belief-Desire-Intention model, we now consider intelligent agents, and specifically agent frameworks which are based on the BDI model where such an approach has been successfully demonstrated. There are a number of agent based architectures which adopt BDI, and we consider the requirements of such an implementation as well as successful applications relevant to our research.

Firstly, to explore some features of the term ‘intelligent agents’ we consider the properties they are expected to have, as put forward by [Wooldridge and Jennings \[1995\]](#):

1. Autonomy: agents operate without direct intervention of humans or others, and have some kind of control over their actions and internal state;
2. Social ability: agents interact with other agents (and possibly humans) via some kind of agent communication language;
3. Reactivity: agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
4. Pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

Most of these features have already been discussed earlier, as we are interested in autonomous agents which are able to perceive and react to the environment and pursue some goal(s). The topic of “social ability” introduces an aspect of communication, but here also makes mention of an “agent communication language”, which raises the question of how agents should communicate between themselves, compared to how communication takes place with entities outside of the agent framework. In [Finin et al. \[1993\]](#) the Knowledge Query and Manipulation Language (KQML) is put forward for agent specific communication as a means not only to communicate information, but to ask other agents to achieve goals. This has been largely replaced by the more recent FIPA (Foundation for Intelligent Physical Agents) standard, supporting a rich variety of message exchange types with BDI-like features (detailed in [Poslad \[2007\]](#)). As discussed in Section 2.1.2, we consider that where a suitable language exists for a components’ own communication (i.e. in this case, FIPA/KQML) then adoption of that specification would be prudent, but where communication to other component types is required, semantically annotated messages add the benefit of including an additional explanatory layer of information around the message, for cases where the recipient may need to reason about what it has been told.

The ability to share plans between agents via a KQML or FIPA mechanism presents a number of benefits. Agents could use such information to better improve the projection element of their SA, and perhaps modify their own plans based on the plans of other agents. In the context of the benefits of plan sharing for goal achievement, [Joyeux et al. \[2008\]](#) has gone some way to implement a plan sharing database for multi-vehicle cooperation in a UAV/UGV context, which raises the question of how such plan communication should be managed. Allowing the free flow of communication between entities could prove beneficial, but it could also cause complete breakdown as entities constantly adapt their plans based on others plan changes. This brings us back to the earlier discussion of communication, that there is a requirement for an appropriate level of information exchange in order to avoid saturation of components and the infrastructure, and allow components to reason about information at a suitable level. There is also an interesting capability introduced by this mechanism, that of one agent requesting another agent to achieve some goal.

## Intelligent architectures demonstrated in the problem domain of SA

There are a number of agent frameworks which have been successfully demonstrated against a number of problem areas, and here we consider some of these architectures where we have found relevance to the SA aspects of our problem choice, which we focus on due to SA being core to a number of the problem statements put forward in Section 1.1.

In the domain of Unmanned Air Vehicle (UAV) control, agent based solutions have been used as an approach for many years. In Karim and Heinze [2005] a comparison of two design approaches for such a scenario is made, one being a typical control systems approach and the other a human cognition approach where the control system behaves in a human-like fashion to “act, sense, behave and decide”. Both implementation approaches in Karim and Heinze [2005] are implemented using JACK (discussed earlier in Section 2.2.1), but it is their second, cognitive BDI approach which has more relevance to our work. In this approach, they draw on the ‘OODA’ (Observe, Orient, Decide, Act) model put forward in Boyd [1996] as a means for reasoning about the environment and appropriate action selection. The ‘Observe’ phase aims to take data inputs and generate beliefs (e.g. sensor feed data stream for position reports to form an xyz belief), ‘Orient’ takes this belief set and combines it with additional contextual information (in this case about the environment), and forms an assessment of how the current situation affects its goal(s). This can be likened to the comprehension phase of SA generation (and perhaps the projection element as well), as there is now some meaning associated with base level perceptions. The ‘Decide’ phase refers to the decision of what to do, or course of action, before moving to the ‘Act’ phase where a plan is put into action. Although no comparison of results from the two approaches (OODA and standard controller) proposed in Karim and Heinze [2005] is provided, their conclusions consider whether combining JACK with cognitive architectures may make validation and verification of software easier. We consider BDI as assisting in the problem statement of Section 1.1 regarding observability of the reasoning process, and draw on the point raised in Karim and Heinze [2005] which suggests it may assist the system designer.

Staying in the UAV domain, Reichel et al. [2008] aim to capture the operational and functional characteristics required for autonomous UAVs, such that the human operator is no longer required. In Reichel et al. [2008], there is consideration of potential cognitive approaches (Soar, ACT-R, BDI and COGNET) which are compared against a number of criteria. It is observed that Soar and ACT-R suffer from over complexity for this type of problem, with a lack of examples in literature of similar applications, and that they are not strong at team formation and coordination. This has relevance to the problem statements introduced in Section 1.1 where topics of appropriate complexity (in knowledge exchange, but of relevance to the intelligence layer as well) and coordination problems are introduced. Furthermore, it is concluded that BDI is more suited to capturing the human operators’ knowledge regarding the UAV operation. We draw confidence in our approach from such work, as it demonstrates the replacement of a human operator with that of an autonomous system, and suggests that a BDI approach can be well suited to such a problem.

In [Ranathunga et al. \[2011\]](#) the Jason ([Bordini et al. \[2007\]](#)) BDI framework is integrated with the Second Life virtual environment, where a football simulation (‘SecondFootball’) is used as the problem domain, recognising events taking place in the simulation and responding accordingly through domain appropriate actions (e.g. turning and kicking the ball). Similarly (in terms of integration), in [Lee et al. \[2013\]](#) Jason is integrated with Second Life, in this case exploring the use of Jason and social reasoning to control polite virtual agents. The demonstration of two approaches from different domains (i.e. football and polite avatar response) using an external environment (Second Life) coupled to the Jason architecture shows promise is being able to control remote entities using this framework.

[Hama et al. \[2011\]](#) choose to use Jason in the UAV domain, providing an alternative BDI approach to the previously described JACK based implemented used in [Karim and Heinze \[2005\]](#). In [Hama et al. \[2011\]](#) the authors seek to integrate Jason with Mikrokopter<sup>1</sup> technology as the UAV component, with a number of (software and hardware) components required to bridge the intelligence layer to the real device. Whereas the previous work discussed ([Ranathunga et al. \[2011\]](#), [Lee et al. \[2013\]](#)) interfaced Jason to a simulated environment, [Hama et al. \[2011\]](#) focusses on real devices, suggesting that Jason may tackle both simulated and realworld devices. This has relevance to the problem statements set out in Section 1.1 concerned with maintaining a general application rather than becoming tied to a specific domain, suggesting that behaviours developed in Jason could have application in both simulated and real domains.

[Fronza \[2008\]](#) also uses Jason, in this case for controlling virtual tanks in an adversarial team context (which we revisit in Section 2.3.3) referred to as TankCoders. This shows Jason agents used to control and coordinate teams of vehicles, with the capability for humans to control tanks as well, providing a mixed human and artificial entity scenario. The software solution in [Fronza \[2008\]](#) also provides a networked capability, allowing multiple instances to be started from different PCs and interact in the same virtual environment. We draw on a number of points from [Fronza \[2008\]](#) which relate to the problem statements in Section 1.1, concerning understanding, observability of the reasoning process, and distributed systems. As the development is itself opensource but also based on opensource components, it could be extended to assess how intelligent agents can communicate to improve each others understanding of the environment (with relevance to the shared situational awareness discussions earlier in Section 2.1.2). Also, as it is based on a distributed architecture, it provides some insight into potential approaches to address that problem statement. To provide distributed communication, [Fronza \[2008\]](#) makes use of the Java Game Networking (JGN), which whilst demonstrated in TankCoders, has a number of drawbacks: it is language specific (Java), no longer seems to be developed, and may lack the wider (i.e. non game specific) functionality needed to support a wider set of domain applications.

In [Settembre et al. \[2008\]](#) the concept of de-centralised cooperation between intelligent agents is proposed, exploring a collective of robots operating in an environment, the issue of how their individual assessments of the environment can be combined, and how collective plan agreement can be achieved. The authors highlight the issue that due to bandwidth limitations

---

<sup>1</sup><http://www.mikrokopter.de/en/home>, accessed September 25th 2014



the collective may be unable to share their individual sensor readings, and instead put forward the approach where each individual puts forward their suggested plan to another robot, and if that robot has no conflicts with the proposal, the plan is forwarded again, creating a form of distributed agreement. This introduces a number of concepts related to the work we consider, firstly raising the question: if a method of communicating at a higher level (i.e. an SA approach) were adopted, would that enable such robots to ratify their understanding of the environment, rather than an agreement of a proposed plan? For example, there may be cases where two robots have an incorrect understanding of the environment with different (mis)perceptions, but agree on the same (inappropriate) plan selection. We consider that the problem statement put forward in Section 1.1 concerned with improving the observability of reasoning and information flow may help address this problem, which we view as a generalised issue, affecting the shared situational awareness (as discussed in Section 2.1) of a collective of artificial entities.

### 2.2.2 Agent organisation

Having examined various approaches regarding intelligent systems capable of addressing the problem areas set out in Section 1.1, we now review work related to the problem statements of resolving problematic situations (i.e. where an agent is experience difficult in achieving a task), and balancing individual pursuits with the wider social welfare. We consider: having given agents autonomy to go and pursue their goals, what happens if two agents are in conflict with each other, or if one agent is selecting actions to the detriment of the wider population? In Hardin [1968] such issues are considered in broader terms, presented as “The Tragedy of the Commons” where individual gain (of an additional sheep owned by a herder, which is then allowed to graze on shared land) is taken due to the perceived small loss to the overall collective. In terms of intelligent agents, for a finite resource, how should an agents’ pursuit of its own goals be balanced with the wider collectives needs for that resource. Following this, we draw on Ostrom [1990] and consider than some organisational structure is required in order to manage agent behaviour.

In Dignum [2003] ‘OperA’ (Organisations per Agents) is put forward as an approach for designing agent societies, comprising an organisational model, social model, and interaction model. Following this design methodology, in scenarios used to investigate the problem statements of Section 1.1 there would need to be a formal definition of roles, interactions, tasks, agent capabilities, social structure, normative behaviour, communication, and so on. Whilst we can see potential benefits in all these areas, of key interest in addressing the problem statement regarding individual pursuit versus wider social welfare is that of normative behaviour, and we explore this further through the use of institutional frameworks.

In defining what we mean by institutions, and how they relate back to the problem areas, we draw on a definition by North [1991]:

*Institutions are the humanly devised constraints that structure political, economic, and social interaction. They consist of both informal constraints (sanctions, taboos, customs, traditions, and codes of conduct), and formal rules (constitutions, laws,*



*property rights*).

Here there are a number of aspects which can be linked back to the topics of problem resolution and managing autonomy. The fact that they can be thought of as a constraint on interaction for both formal and informal rules could be used to address both areas of the problem. Firstly, if an intelligent agent has no ‘onboard’ directive for what to do in a given situation, it could receive guidance from the institution. Similarly, in the management of autonomy problem, we consider the use of institutions as a mechanism to curtail an agents freewill should the need arise, furthermore with the flexibility of dynamically changing the institution response (e.g. depending on environmental conditions, economic conditions, political, etc), in order to address the over pursuit of self goals whilst neglecting the wider society scenario put forward in [Hardin \[1968\]](#).

In [Cliffe et al. \[2006\]](#) institutions are formally defined by a mathematical model and implemented via the translation of ‘InstAL’ (Institutional Actional Language) to an Answer Set Programs. An Answer Set Solver takes this program and a set of events and produces a set of (institutional) facts, which can be interpreted as guidance from the institution (a more detailed overview of this process is provided in Section 4.1.4). For our problem domain, we can consider the case where an individual entity is unsure of what action is required or expected, and could be issued with guidance from the institution to assist. Typically this might take the form of an obligation to perform some action, however its worth noting that obligations are not necessarily mandatory (though some penalty for their violation can be implemented). Given the problem areas put forward in Section 1.1 of resolving difficult situations, and managing autonomy, this presents some helpful capability. In the category of ‘difficult situations’ we could include cases where an entity does not know what to do, and also cases where one entity is in conflict with another (e.g. a deadlock situation with both entities trying to access the same resource or perform some exclusive, blocking action). For the former, institutions could play a role in providing guidance as to what action is appropriate, and for the latter could issue obligations to both parties in order to resolve the conflict.

The Jason agent platform (discussed earlier Section 2.2.1) has been demonstrated in conjunction with the InstAL framework, as a means of regulating agent behaviour. In [Balke et al. \[2011\]](#) this approach is explored in the context of Wireless Mobile Grids (WMG), in a scenario seeking to address the problem of ‘freeloading’ rather than paying a small (energy) cost. This work has additional relevance, as it is considering a technology (WMG) which is not yet mainstream, i.e. of the near-future disruptive technology type considered in Chapter 1. The more general concept of communication management is also discussed in the further work of Section 7.3.1, to address behaviour of members within the framework proposed in this work.

Further work of relevance is presented in [Lee et al. \[2013\]](#), where Jason agents are coupled to the Second Life virtual environment and integrated with InstAL institutions. In this case, the topic of interest is how to generate polite behaviour in virtual characters, through the application of social convention as obligations directed at Jason agents controlling non-player characters. Of additional relevance in this work is the adoption of a shared communication

layer between the virtual environment, the Jason platform, and the governance framework, and this is discussed further in Section 2.2.3

In other work, Bradshaw et al. [2004] touch on the notion of potential actions vs. permitted actions, and the use of their ‘KAoS’ framework to specify how an agent should be constrained. Their discussion of adjustable autonomy is also relevant to the research presented here in two ways. Firstly, there is likely to be a requirement to vary the level of autonomy the vehicle has, such that it is compliant with relevant governance systems. Secondly, there is also the need to adjust its autonomy depending on the situation the vehicle is in, i.e. in a convoy vs. travelling alone.

Campos et al. [2010] propose an ‘assistance layer’ to manage the organisation of agents. This uses norms to influence agent behaviour, and provides a case study looking at Peer-to-Peer (P2P) networking, with norms used to manage bandwidth use between peers (i.e. considering Hardin [1968] the commons is represented as available bandwidth). Additional information seems to feed from knowledge of ‘environment observable properties’ e.g. available bandwidth and latencies specific to each P2P’er, so the norm agent role is both as mediator and as rule enforcer. Intuitively this does suggest a substantial increase in the complexity of both the system and its behaviour, and Campos et al. [2010] go onto make a similar point, that changing norms will have an impact on organisational change and system outcomes, though this may be desirable.

These examples of regulating agent behaving through the use of norms provide support for the view that there is merit in such an approach, and that it can offer an improvement over ‘unregulated’ behaviour. Furthermore, the addition of such a framework may bring further benefits in assisting agents in situations where the agents can find no suitable action through their own reasoning.

### 2.2.3 Communication frameworks

The topic of communication has been discussed earlier in Section 2.1.2 in terms of information exchange between entities, and how this is required to improve the collective shared situational awareness. There was also some discussion of agent communication approaches (KQML and FIPA), but now we consider approaches related to broader communication. In Section 2.1.2 there was the introduction of the concept of communication ‘channels’ as a means to enable entities to choose what information they have an interest in, and a need for the information to be ‘semantically rich’ in order to assist the SA generation based on received messages. Here we consider what technologies could assist in support such an approach.

The desire to exchange information containing more than just the data itself (i.e. adding semantic annotation to provide an explanation of the data being sent) has generated a number of research efforts, and we consider the World Wide Web Consortium (W3C) specification of the Resource Description Framework (RDF) as a data model of interest. Information in this specification is of the format subject-predicate-object (referred to as a ‘triple’), with the possibility of one RDF message comprising of multiple triples. In Section 2.1.2 we discussed a

Listing 2.1: An RDF example structure and content

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sensors="http://127.0.0.1/sensors#">

  <rdf:Description rdf:about="http://127.0.0.1/vehicleTypes/Lorry">
    <sensors:color rdf:resource="http://127.0.0.1/colours#red"/>
    <sensors:speed>30</sensors:speed>
    <sensors:takenAt>1402738172</sensors:takenAt>
  </rdf:Description>

</rdf:RDF>
```

desire to add descriptions around data such as unit definitions, sensor names, creation times, in order to add richness to communicated data, and we explore RDF as an option for doing so.

An example of what such a reading could look like is shown in Listing 2.1, which describes a message about a vehicle. In this example, the subject is `http://127.0.0.1/vehicleTypes/Lorry`, and there are predicates of colour, speed and the time the sensor reading was taken at, with corresponding values being the objects. Of note is the URL format of these definitions, which makes it possible to provide shared definitions of the terms used, e.g. `http://127.0.0.1/colours#red` could be queried to understand what is meant by the colour red. Furthermore, there are query languages (e.g. SPARQL) which allow such information to be examined further, with the example of the earlier vehicle RDF, this could be to find all vehicles travelling above a certain speed. Combining this with the availability of other reference sources (e.g. with the vehicle example sources such as <http://data.gov.uk/dataset/gb-road-traffic-counts> provides road and traffic count information), this adds significant support for intelligent systems to reason about received data, with the possibility of raising their SA.

Following this, as well as considering *what* is communicated, we also need to consider *how*. The demand for communication between one component and another has led to the development of a vast array of approaches and middleware architectures to support communication. The High Level Architecture (HLA) has been demonstrated in a wide variety of such applications in computer based simulations, commonly used in military training simulations (e.g. Dariusz Pierzchała [2011]) but also in multi-agent simulations using commercial games engines (e.g. Plch et al. [2012]). HLA has been established as a robust framework, however issues exist over the provision of the ‘Run Time Infrastructure’ (RTI) component where open-source implementations have historically lagged behind licensed alternatives. Whilst the equally established IEEE-specified Distributed Interactive Simulation (DIS) protocol exists for coupling simulation systems, HLA provides additional levels of simulation management with the key extension of time management (Fujimoto [1998]). However, whilst the work presented in this thesis happens

to use the example of a simulated problem domain as a test case, it is a requirement too that it remain applicable to real world situations as well, which provides some justification for not adopting simulation specific approaches such as HLA and DIS.

In [Fritsch and Wrede \[2007\]](#) the XCF framework is adopted as an approach of XML messaging for distributed systems, used for interactive robots and promoted for its simplicity, suitability for rapid prototyping and modular based approach. Their framework SDK comprises of an API for XML processing, the XML Communication Framework (XCF) itself, an ‘Active Memory XML Server’ for coordination and event management, and a ‘Petri-Net’ engine to provide some additional coordination. Having introduced the framework as simple, this seems to rapidly add complexity to their approach, negating the benefit of a simpler XML approach and instead requiring message schemas to be agreed, as well as complexity around the process. However, this is likely to be the case when a framework includes the provision of the infrastructure components, i.e. a messaging server with associated memory managed, offline message storage, and so on. [Hawes and Hanheide \[2010\]](#) presents another approach to distributed robot communication, where components subscribe to “working memories” with the ability to read, add, delete and overwrite information stored in working memories, and an event notification system for when these memories are updated. Such an approach has some similarities to a publish-subscribe mechanism such as that in XMPP, though in [Hawes and Hanheide \[2010\]](#) this is replaced with the interesting concept of representing the ‘memory’ of an object value within the framework (in XMPP terms, on the node). Whilst this could potentially offer a framework to manage the shared situational awareness of an environment, it would seem to limit the ability to situate the intelligence layer directly into the environment, and allow agents to update their own belief set. Both [Fritsch and Wrede \[2007\]](#) and [Hawes and Hanheide \[2010\]](#) are based on the ICE middleware framework [Inc \[2005\]](#) – in itself a strong middleware component – along with other potential solutions such as the use of Enterprise Service Bus (also demonstrated with application to intelligent agents in [Cranefield and Ranathunga \[2013\]](#)). Whilst there are clearly a variety of potential solutions to message exchange, we seek to retain focus on the information itself, rather than be drawn into an exhaustive comparison of architectures. From the desire put forward to exchange semantically rich information, RDF seems an appropriate message format, and the ability of clients to ‘fire and forget’ the message as enabled by the publish subscribe approach provides a good match to our requirements. Whilst other approaches may offer their own strengths, the simplicity of this design is appealing in keeping the implementation lightweight, compared to the complexity of the alternatives, e.g. defining HLA federations, managing ICE slice definitions, or complex configuration of messaging frameworks. Instead it gives us a straightforward, generalised approach applicable to both simulated and real entities, which we feel appropriate to support our research.

In [Behrens et al. \[2011\]](#) the topic of a general interface for agent platforms to environments is considered, with the proposal of an Environment Interface Standard (EIS), which would relate to the problem statement of Section 1.1 regarding a distributed systems approach. FIPA is considered as an alternative, with EIS proposed as a leaner option, capable of connecting to virtual environments and thus support a wider range of integration, with bridges already

developed for a number of agent platforms. Discussions are presented based on examples of the number of interfaces which have been built to Unreal Tournament, regarding the duplication of effort in writing various interfaces which all provide similar functionality. Whilst we agree in the goal of abstraction, and reducing code re-use, there is a question of where this abstraction should occur. A generalised, semantically annotated message approach provides the benefit that the message format is known and accessible, but each component has to handle that information appropriately (although the effort is mitigated by technical issues such as (de)-serialisation being handled by existing libraries). The EIS approach removes this issue, but raises the question of how customised handling could be developed, e.g. a level of pre-processing of messages (if required) before agents are updated with environment data.

## 2.3 Problem Domain

The problem questions discussed earlier in the introduction (see Section 1.1) are intentionally high level, and applicable to the general rather than the specific. As new technology, empowered by intelligent autonomous agents, enters and interacts with the mainstream human population, we seek to identify what approaches from computer science can aid its understanding of the environment, its ability to pursue some goal(s) and in doing so resolve situations which are unhandled by onboard plans, and those requiring coordination with other human and autonomous entities.

However, we also seek to ground this investigation in some real world problem domain. The reason for this is twofold: firstly to demonstrate the application of the work in solving a ‘real’ problem, and secondly to more exhaustively test our proposed architecture (and in doing so, identifying issues which had not been initially considered). With this motivation, we elected to explore in the broad area of intelligent transportation systems, and more specifically autonomous vehicles.

In discussions of the problem questions earlier, we stated our interest in near-future technologies which are likely to be disruptive due to their innovation and impact on how things are currently done. The area of intelligent vehicles has seen a significant increase in technological developments over the last few years, from driver assistance devices through to demonstrations of fully autonomous vehicles, and although not a common sight on the roads just yet, that time seems to be coming ever closer.

As the work presented in this thesis does not need access to such vehicles, because our focus is in more high level issues around coordination and knowledge communication. In consequence, it is quite adequate to replace real life devices with simulated vehicles and traffic. This has the advantage of allowing potentially dangerous scenarios to be explored, in a repeatable fashion.

In order for this effort to be useful, a reasonably broad review of related work in the area was required, including intelligent vehicles, vehicle to vehicle communication, and the simulation of both individual entities and traffic populations. This is to avoid over simplifying the problem, negating the expected benefits of testing against this domain, and also to inform our work on what real life capabilities and limitations may exist in such a technological deployment.

### 2.3.1 Intelligent Vehicles

We first consider recent developments in the area of intelligent vehicles, not because we are looking to directly interface to such vehicles, but rather to be informed of what such capabilities may look like, what they may enable, and how we can adopt a generic enough approach which would still be capable of interfacing with such vehicles.

Early progress in this field was driven by the DARPA series of ‘Grand Challenges’. These events aimed at pushing the art of the possible in order to advance capability in this area. The first challenge in 2004 did not achieve significant success, but the following 2005 and 2007 events achieved impressive goals of navigation through unknown routes, challenging terrain and operating in urban environments.

The 2005 competition was won by the ‘Stanley’ (see [Hickey \[2005\]](#)) entry from Stanford, based on a modified Volkswagen Touareg. There are novel approaches adopted in their approach, discussed in [Davis \[2006\]](#), where developing awareness and reasoning about the vehicles own perceptions has proved the most fruitful, and this improved reasoning is put forward as being able to offer improved handling of sensor noise. We consider that reasoning about an entities own perceptions can be linked back to topics discussed in Section 2.1.1 regarding SA, and the potential benefits in questioning one’s own SA (reflection), suggesting that this would be a useful capability for a framework to provide.

The winner of the 2007 competition was the ‘Tartan Racing’ entry from Carnegie Mellon. Focusing on the AI implementation of their solution as discussed in [Urmson et al. \[2007\]](#), the “Behaviour Generation” engine has been implemented as a state machine, using high level behaviours (with three contexts: drive-down-road, handle intersection, and achieve-zone-pose for parking) which then have supporting sub-behaviours. Their “Motion Planning” aims to avoid collisions with static and dynamic objects, and seems to be concerned primarily with road following. The “Perception and World Modelling” seems to comprise a data fusion engine, driven by a map of both static and dynamic obstacles. We reflect that an approach of breaking required functionality into components, with an appropriate level of knowledge representation, proved an effective solution.

Within Tartan Racing’s [\[Urmson et al., 2007\]](#) “Perception and World Modelling” component substantial data fusion takes place, with base level sensor feeds being fused and classified to provide some “Situation Assessment” (their term). In the fusion process an object is classified into a more explicit type, if possible, (e.g. change a set of detected points into a ‘car type’ if it meets the requirements of being a ‘car’), and we note this in relation to discussions earlier in Section 2.1.2 regarding data fusion, as a working demonstration of how to improve SA. Regarding “Situation Assessment”, it is stated that the “layer attempts to estimate the ‘intention’ of the tracked object by integrating the estimates with knowledge about the road world model”, which bears a resemblance to the SA discussions earlier in Section 2.1.1. It is also reported that the system struggles to perform well when approaching intersections and projecting future events (e.g. whether a vehicle will leave or join at that intersection). This is another example of how communication of higher level information between vehicles *could*

prove useful; rather than having to rely on some visual cue (e.g. an indicator light), vehicles would have been informed as to what was likely to happen at that intersection based on other vehicles exchange of future plans. As well as having the benefit of aiding autonomous vehicles in working together, there is the additional benefit that excessive braking and acceleration are reduced as vehicles are able to predict events rather than relying on last minute reactions.

Further discussion relating to the ‘Tartan Racing’ entry is provided in [Urmson and Whitaker \[2008\]](#), showing details of potential benefits of automated vehicles. Worth noting is the limitations placed on the scenario that this vehicle was used in; no pedestrians were present, traffic lights were limited to red only and at pre-defined points. We draw on this as a reminder, that if an overly complex scenario is used then our framework may fail; we need to consider a balance between over simplifying the problem domain versus adding too much complexity.

Two individuals involved in ‘Grand Challenge’ entries, Sebastian Thrun and Chris Urmson, went on to become involved in the Google Car (e.g. discussed in article [Markoff \[2010\]](#)), resulting in a fleet of Toyotas which have now covered (as at 2012) more than 190,000 miles (figure from [Guizzo \[2011\]](#)), and a recent blog post by Urmson ([Urmson \[2014\]](#)) stating that as of 2014 the have now logged 700,000 miles. Setting out to achieve the goals of “Reducing road accidents, congestion, and fuel consumption” provides us with some indication of potential challenges we could seek to address in our test scenarios, which are considered in Chapter 5.

Other manufacturers are making similar developments, with recent announcements such as Nissan’s ([Gordon-Bloomfield \[2013\]](#)) further demonstrating the spread of such capability. We draw on this to increase confidence that assumptions regarding technological capabilities made in this thesis are close to (if not already) in existence. Physical sensors are capable of providing increasingly accurate geospatial and environment information, to an intelligent control system, capable of generating appropriate commands in order for the vehicle to achieve its goals. With such individual vehicle capability emerging, a question follows as to what communication between vehicles would be possible (with infrastructure capability considered in Section 2.3.2) and what information could exchanged for individual or group benefit (in general terms considered earlier in Section 2.1.2).

Considering alternative autonomous transportation, the development of Urban Light Transit (ULTra) presented in [Lowson \[2003\]](#) as a means of vehicle transport has been successfully in place in London’s Heathrow Airport since 2011. In this approach, light weight vehicles have their own guide way constructed (simplifying the navigation issue) with vehicles continuously moving around the route. The overall network is considered as a number of ‘slots’, with each vehicle then filling a ‘slot’ with an appropriate gap ahead based on speed and braking. Vehicles are able to merge back into these ‘slots’ on picking up passengers and continue along the route. An ‘empty vehicle management’ system is used to address the availability of vehicles in order to reduce wait times, planning vehicle availability around demand by managing where to park vehicles when not in use, and minimising delivery time to pick up points. This demonstration of a level of coordination between autonomous vehicles is taken as a positive indication of what is possible, and although the traffic network operates segregated from human drivers, it is still dealing with the challenging problem of controlling real world vehicles, handling their sensor



data, and planning future manoeuvres.

### 2.3.2 Vehicle communication

Section 2.1.2 introduced the topic of communication, regarding what information could be exchanged and how this could help understanding, and messaging frameworks were discussed in Section 2.2.3. Here we consider communication further with regards to capability in the chosen problem domain of vehicles, in order to align our experimentation with real world developments and base our communication requirements on likely real world capability. However, it is not our intention to adopt the limitations of current technology; we aim for a general approach to the problems set out in Section 1.1 and so we do not specifically limit ourselves to the performance of current solutions (which will almost certainly get better over time) in a specific domain (because we wish to be as domain-agnostic as we reasonably can).

Communication between vehicles is referred to variously as Vehicle to Vehicle (V2V) and Vehicular ad hoc networks (VANETs), considered as a subset of the Mobile adhoc network (MANET) family. Developments in this area have been supported by availability of required technology (e.g. emerging IEEE 802.11 standards and routing protocols such as those reviewed by [Altayeb and Mahgoub \[2013\]](#)) but also as vehicles become able to measure and capture useful information (e.g. sudden prolonged use of brakes) which would be of benefit for other vehicles to be made aware of. The US National Highway Traffic Safety Administration (NHTSA) announcement [[Naylor, 2014](#)] that Vehicle to Vehicle (V2V) communication devices may become mandatory in a year adds some weight to the assumption that such communication will be seen increasingly in the future. Another development is that of ‘Floating Car Data’, where information such as vehicle speed, location, and direction is reported back to some other system, in order to assist with activities such as congestion monitoring. Such information can be extracted from the vehicle itself as detailed in [Huber et al. \[1999\]](#), or alternatively via a mobile phone onboard the vehicle. We have an interest in such developments, as it indicates both the near availability of rich data and also some domain specific benefits (collision avoidance, congestion reduction) where such information can be of benefit.

Such communication capability can also be used to support coordination between vehicles, with one such example being that of vehicle platooning, or vehicle convoys. [Bergenheim et al. \[2010\]](#) investigates the use of autonomous vehicle controls to enable ‘vehicle platoons’. This work is part of the EC-funded “SAfe Road TRains for the Environment” (SARTRE) project, whose aim is to develop the capability to allow a number of autonomously controlled vehicles to follow one human-driven vehicle. Publications from this project identify the required functionality to control convoys, as well as some consideration for what may be communicated (control and coordination of the platoon). This work also provides examples of the potential benefits of vehicle platoons: up to twenty percent reduction in fuel consumption, ten percent reduction in fatalities, and improved driver convenience (for passenger-drivers in the vehicles where control has been ceded to the platoon). Further benefits have also been discussed in [Dressler et al. \[2011\]](#), where improving traffic efficiency is a key goal.



Widening the scope from vehicle to vehicle communication, there is ongoing research in vehicle to infrastructure (V2X, V2I), which also has the potential to enhance individual SA (e.g. a vehicle is informed of an upcoming traffic light state) as well as group SA (e.g. all vehicles receive the same, accurate, information). Kim [2012] has adopted such an approach, and demonstrates benefits in implementing communication between traffic lights and vehicles, in order to improve fuel consumption and reduce emissions. Similarly, Audi (published via parent company website Volkswagen [2013]) have demonstrated vehicles retrofitted with a device to allow them to interact with traffic lights, that appears to show real benefits for traffic flow. Specific improvements cited include a reduction of CO<sub>2</sub> emissions by up to 15 percent, and substantial (sic) fuel savings, but the lack of precise details makes verification difficult.

In Leontiadis et al. [2011] the total trip time for journeys is shown to be significantly improved through vehicle to vehicle communication. This study shows that if navigation systems share traffic information, then journey times are shortened, and we draw on this, as well as the previous examples to conclude there are significant potential benefits from improved communication in the vehicle domain. We suggest this may be due to both improving the collective SA, but also as it enables better coordination (considering congestion and managing arrival times at traffic lights in the sense of a coordination issue). As these topics relate to the problem statements put forward in Section 1.1, it builds confidence in the view that this is a suitable problem domain in which to operate.

Finally, to inform our work in the area of how much communication may be possible based on such technology, the capability of such networks needs to be assessed, in terms of metrics such as transfer speeds, latencies, packet loss etc. There is an inherent difficulty in the physical nature of establishing a moving networks (e.g. Doppler effect due to speed as considered by Albarazi et al. [2011]), as well as the impact of environmental considerations (e.g. urban environments, vehicle congestion). As such, it seems reasonable to expect performance to be lower than that of a stationary 802.11b/g/n wireless network. In Jiang et al. [2008] a transmission speed of 6 Mbps is considered, however alternative communication strategies such as cellular approaches (e.g. Li et al. [2012]) may offer improvements on this. In short, this is a rapidly developing area, and what communication performance is available now may be considerably different in a few years time.

### 2.3.3 Entity Simulation

This section considers various aspects of simulation, with focus on the vehicle problem domain, but also related areas where intelligent agents are being used. Discussion is included of developments in physics modelling, for two reasons. Firstly that if we are building intelligent agents, there is a question of whether they should control an individually simulated entity, and if so whether there is sufficient fidelity in available software packages to do so. Secondly, that in line with our desire to remain applicable to multiple domains, it is of interest if there are suitable modelling packages which would support the use of our framework in another application area.

Firstly we consider the the RoboCup competition (Kitano et al. [1997]), for its relation to

the problem domain of this thesis, as issues of goal pursuit, SA, coordination and cooperation all occur. First held at IJCAI-97, it sought to establish a benchmark AI problem event where due to the nature of the competition, emphasis arises on team, as well as individual, performance. A team of robots are required to interact together, in order to defend against an opponent, and to attack in order to score a goal. Such a competition has multiple areas in common with the work presented in this thesis, with the multi agent approaches, the issue of team situational awareness, and it could be considered a multiple autonomous entity coordination problem.

In [Mitrović et al. \[2013\]](#), the RoboCup simulator (SimSpark) is used to assess the use of Jason (as discussed earlier in Section 2.2.1) agents at performing required tasks (searching for the ball, walking to the ball, kicking the ball, scoring) for this challenge. This work discusses the performance requirements of such an integration, where SimSpark requires agents to complete their reasoning cycle within a 20ms limit, and goes on to show that on modest hardware, it is possible to run six agents from a single PC. Also of interest is the discussion on the use of JaCaMo (as presented by [Boissier et al. \[2013\]](#)), a framework consisting of Jason, ‘Cartago’ as an application for programming the environmental components, and ‘Moise’ as a model for the organisation specification. Following on from the discussions in Section 2.2.2 regarding organisational frameworks, work put forward in [Hubner et al. \[2007\]](#) proposes the integration of Moise with the Jason BDI framework to manage the organisational aspects of a soccer team. Of interest to the problem statements we introduced in Section 1.1, is the Jason framework being extended to provide organisational aspects, and also the use of a domain specific example taken in [Hubner et al. \[2007\]](#) to explore and highlight their approach through specifics, whilst retaining a general solution, aligns with our desire to retain application outside of the domain of intelligent vehicles.

In [Fronza \[2008\]](#) an adversarial game context using intelligent vehicles is presented, based on a networked multi agent system, of relevance to the problem statement of Section 1.1 regarding distributed systems, and this is used in our initial prototyping work detailed later in Section 4.1.3. In the system discussed in [Fronza \[2008\]](#), either humans or Jason agents are able to control a team of tanks, with the goal of defeating the rival team of tanks. This work integrates the jMonkeyEngine 3D Java Engine with the Jason platform, with jMonkeyEngine modelling vehicle physics and response to agent requests, and updating the belief set of Jason agents based on key events such as being hit by an enemy tank, detecting an enemy tank. Whilst the approach taken in TankCoders contains coupled components (i.e. the agent platform, vehicle modelling, and 3D representation are all instantiated together) it does introduce networked functionality, where additional players can join and interact in the same environment. Considering the problem statements introduced in Section 1.1, this capability addresses numerous areas, and in general terms demonstrates an intelligence layer managing a society of entities to bring about some goal. We discuss the effort undertaken to introduce a distributed systems approach, exchanging semantically annotated messages, later in Section 4.1.3, where the solution of [Fronza \[2008\]](#) is modified to address the remaining problem statements regarding observability of message exchange.

Similar approaches can be found in dTank [[Morgan et al., 2005](#)], again using an adversarial

agent tank framework but in this case, coupled with both the Java Expert System Shell (JESS, mentioned earlier in Section 2.2.1) rule engine and the Soar cognitive architecture. Later work by Ritter et al. [2007] improves the richness of the simulated environment, whilst still maintaining a similar level of self-awareness (location, speed, available ammunition) communicated to the cognitive layer. An interesting point is made that additional information is needed to be passed to the agents to allow more realistic reactions, with the example of being informed of nearby shots compared to the existing information exchange (e.g. of being hit). This needs to be considered when operating with sensor feeds to agents, that unless there is some environment perception capability (i.e. a vision library, audio processing) then agents are only able to reason about the (potentially narrow set of) information which the system passes to them. In the example here, agents were not told about nearby shots, and so expected behaviour such as run, or take shelter, did not occur because the triggering events were not perceived and so did not contribute to individual/group SA. This issue relates back to concepts put forward earlier in Section 2.1.2, and in general terms the case that agents may need assistance in message exchange, to avoid too little or too much communication, and in finding additional information about a particular subject, which is discussed in the further work of Section 7.3.1.

Moving to consideration of simulation fidelity, Craighead et al. [2007] conducts a survey of a variety of unmanned vehicle simulators, which overlaps the area of interest covered by the work presented here. This survey considers commercial game engines (including Unreal engine, FarCry, X-Plane), the MATLAB package, and open source simulators (of note being SimRobot for its use in the RoboCup competition as discussed in Section 2.3.1). Craighead et al. [2007] concludes that there is no longer a need to build custom (robotic) simulators from scratch in order to achieve a high level of fidelity, and that open source simulators are now of sufficient quality to compete with commercial alternatives. As we seek to address the problem statements of Section 1.1 we need to represent suitability complexity in the problem domain in order to challenge the intelligence layer, and in Section 4.1.3 we discuss that alternative vehicle representation is needed. When combined with the distributed component based approach set out in the problem statements, the findings of Craighead et al. [2007] suggest we can draw on open source components to inject additional entities into the scenario if required.

A further area of consideration is that of physics modelling packages, which create the possibility of defining a simulated entity and applying real world physics to model the entity's response (e.g. the approach in TankCoders described earlier). A review of a number of such packages is presented in Boeing and Bräunl [2007], concluding that the physics engines (AGEIA PhysX, Bullet, JigLib, Newton, Open Dynamics Engine, Tokamak, and True Axis) were all suitable for game development. Whilst the author of Boeing and Bräunl [2007] has written a "Physics Abstraction Layer" to provide an abstracted interface to these physics engines, it still leaves the issue of requiring a bottom up creation of some simulation, and whilst Boeing and Bräunl [2007] states all the physics modelling within the various engines is satisfactory for their test cases, these are all quite low level (i.e. static friction, bounce heights). Where this would be of relevance to those involved in high fidelity simulation, we are more interested in an 'appropriate' level of domain representation in order to investigate problem statements

in general terms. However, the inclusion of Bullet in the review is of interest, as the Java port of this (JBullet) is used in the jMonkeyEngine platform, part of our solution discussed in Chapter 4. This physics model was also used in the TankCoders work discussed earlier, and so offers confidence that the physics involved in the area of vehicle simulation can be implemented credibly based on this package.

The openDS application [Math et al., 2012] is an alternative approach, using the jMonkeyEngine framework and focussing on vehicle simulation (making use of the JBullet physics engine described in 2.3.3). Further developments are presented by Vokrinek et al. [2014] which have added an intelligent agent layer based on Alite (an agent orientated toolkit presented in Komenda et al. [2013])) to explore both human in the loop and autonomous vehicle behaviour. However, this work seems focussed on the individual human in the loop, where traffic representation is achieved through agent representation based on Alite, rather than drawing on an existing proven traffic simulation package. If the intention is therefore to explore the what-if of intelligent agent vehicles on the road, then representing these agents in an existing proven agent architecture may help remove uncertainty regarding unexpected behaviour (e.g. is it the agent, or is it a bug in the architecture) as well as providing a richer set of capabilities (e.g. such as Moise integrated with Jason discussed earlier in Section 2.3.3).

#### 2.3.4 Traffic Simulations

For the problem domain of intelligent vehicles, there is a question of whether to adopt the approach of multiple instances of individual vehicle simulations (e.g. using physics models such as those mentioned previously in Section 2.3.3), or to use large scale traffic simulations. Individual entity modelling may offer greater fidelity of a single vehicle but lack the ruleset and know how to interact (e.g. road following, turn precedence, etc) whereas a traffic simulator is likely to introduce these features, but may not offer the capability for external control over a specific vehicle (e.g. for the intelligence layer to take control of a member of the traffic population). In order to assess how intelligent vehicles behave in more realistic conditions, traffic simulators have been explored to provide both rule behaviour for individual vehicles (e.g. road following models) and additional vehicles with which our agent controlled vehicles have to interact. Further more, these simulators are capable of providing additional environmental features such as traffic lights, speed restrictions, queues, and so on. For our generalised problem topics this is highly desirable, as we are in effect ‘bolting on’ our intelligent agents to a much richer environment, where issues such as coordination, conflict, and dealing with unexpected events are more likely to arise, both in planned specific scenario to investigate such topics but also unintentionally as a (desirable) by product of the simulation complexity.

Large scale traffic simulations are generally considered in terms of being microscopic (e.g. vehicles modelled at an individual level per simulation step) or macroscopic (e.g. higher aggregated behaviour to represent flows). As the work here is interested in intelligent control of individual vehicles, as well as their interactions with other vehicles, we have only considered microscopic traffic simulators. Furthermore, we are likely to require extracts of individual ve-

hicle characteristics and each time step in order to assist with SA generation and to provide appropriate stimuli to agents.

There are a number of software packages which focus on traffic simulation at such a microscopic level. MATSIM (e.g. presented by [Balmer et al. \[2008\]](#)) is one such simulator, a well established Java based package, using an agent representation at the individual vehicle level. Agents contain an XML definition of their ‘plan’, defining routes, journey times, and in fulfilling this plan create the traffic simulation. In considering potential performance requirements, [Waraich et al. \[2009\]](#) presents the scalability of MATSIM with 1 million roads and 7.3 million agents based in Switzerland, demonstrating large scale multi agent simulation.

The SUMO software (see [Krajzewicz et al. \[2012\]](#)) is also explored as a potential package for providing the traffic simulation component of the research. SUMO is a microscopic traffic simulator, capable of running at various timesteps in order to provide real-time or faster than real-time simulations. Available as open-source software since 2002, the package provides the capability to generate road networks and generate traffic flows from a variety of source data. An interface is provided (referred to as TraCI, Traffic Controller Interface) which adopts a TCP/IP based implementation, accepting requests via specific data packet constructions to provide the capability of extracting simulation data, and also controlling various aspects of the simulation. A number of methods have been developed to interface using this functionality, namely Java (Traci4J<sup>2</sup>) and Python, creating the possibility to extract required information from the simulation, and communicate it to the wider simulation participants (i.e. the intelligent agents).

SUMO has been demonstrated in a number of relevant areas to the research presented in this thesis. The work of [Soares et al. \[2013\]](#) presents SUMO integrated with the JADE agent framework through the use of the TrasMAPI (Traffic Simulation Manager Application Programming Interface) API. This is an interesting approach as it attempts to provide a level of abstraction from SUMO specifics, to non SUMO specific traffic terms. Relating to the discussion in Section 2.2.3, this is a positive sign, as we wish to develop this even further, to the more generalised semantic message based approach.

Other related interface work in SUMO is that of [Grumert and Tapani \[2013\]](#), where a Python application is developed using the TraCI interface, to control a cooperative Variable Speed Limit (VSL) system. The approach taken in this work is to communicate upcoming speed restrictions via Infrastructure to Vehicle (I2V) rather than relying on upcoming gantry displays. The scenario also makes use of ‘detectors’ (sensors placed in the SUMO simulation which can extract passing vehicle information) to measure traffic speed, which is used to calculate an appropriate speed limit restriction. The premise is this allows vehicles to be informed of upcoming speed restrictions earlier, and to receive individual limits based on speed and position.

These two research efforts ([Soares et al. \[2013\]](#), [Grumert and Tapani \[2013\]](#)) are of particularly of relevance to our work, as they show the ability to modify vehicle behaviour based on measured parameters from the scenario, and that the overall simulation can be integrated with an agent platform (suggesting the feasibility of agents controlling vehicles based on perceived

---

<sup>2</sup><https://github.com/egueli/TraCI4J>, retrieved 25th August 2014

values from the simulation). The investigation into improved management of variable speed limits presented in Grumert and Tapani [2013] is of direct relevance to one of the experimental scenarios presented in this thesis (see Section 5.6), suggesting promise for that particular scenario. However, to address the full problem statements set out in Section 1.1, we need to develop a more generalised approach, and whilst the adoption of some abstraction in Soares et al. [2013] removes dependency on particular traffic simulators, it is still domain specific. Furthermore, we seek to ensure a more observable communication approach, where the adoption of semantically annotated messages assists both in developing a generalised solution, but also in supporting the introduction of additional components into the framework as required (e.g. governance structures).

Whilst there are alternative vehicle simulation packages, along with interfaces to such packages and those discussed earlier in this section, they are not presented in more detail here. From the problem statements of Section 1.1, as intelligent vehicles are the problem domain chosen to investigate these statements, then we require the chosen simulation software to provide a number of capabilities. A more in depth discussion of this follows in Section 4.1.3, but key amongst these requirements is the ability for an intelligent architecture to control individual vehicles from within the simulation, and extract performance measures of that vehicle, as well as the wider vehicle population. This simulator needs to provide this in realtime, as we assume both the agent framework and other components will operate at that speed, and we seek a simulation tool which has been demonstrated in scenarios where sufficient complexity will meaningfully explore the problem statement of Section 1.1. Whilst there may be alternative packages available, SUMO seems to meet these requirements, but equally if an alternative becomes desirable then migration should be straightforward, since our problem statements expressly state we need to demonstrate a generalised approach.

## 2.4 Summary

This chapter presents a consideration of three key research areas with respect to our problem questions. Human factors were considered first in order to set the scene for our meaning of the understanding, communication and situational awareness (SA). We consider ‘low’ and ‘high’ SA and the desire to communicate associated knowledge appropriately to improve the overall shared situational awareness.

We then consider computational issues, with focus on intelligent systems suitable for investigation of the problem statements from Section 1.1. We consider a variety of approaches, from the more heavy weight cognitive architectures such as SOAR and ACT-R, to focussing on the Belief-Desire-Intention (BDI) model. The BDI approach is explored as its roots in folk-psychology and use of human-like terms may assist in the problem statement regarding observability of the reasoning process by allowing other BSF components access to the ‘mind state’ of agents (which can then be presented to a human debugging the system). A number of research efforts are discussed where BDI has been demonstrated successfully in UAV and individual entity control (i.e. football players, virtual avatars, virtual tanks), suggesting promise in

its use for this genre of problem. Governance of such agents is considered via the use of institutional frameworks and the issuance of norms to offer agents guidance in situations where they have no clear directive, or are unable to resolve the situation themselves, and we seek to use such capability in order to address the problem statement of Section 1.1 concerned with dealing with difficult situations. We also consider research related to Wireless Mobile Grids, where institutions have been shown to address the “tragedy of the commons” type of issue, where individual pursuit of self goals needs to be regulated to ensure fairness to the wider collective, which addresses the related problem statement of Section 1.1. We also discuss communication frameworks which would allow such agents to communicate both between themselves (such as KQML and FIPA) but more generally via a publish-subscribe approach of XML based RDF messages, considering the problem statements of Section 1.1 concerned with communication and improving the observability of exchanged information. We consider that whilst specific communication approaches may be suitable for use internally within components (e.g. FIPA for the intelligence layer), that external communication should be based on accessible, semantically annotated messages, to offer recipients improved capability to reason about that data, and derive an understanding of what the data means.

Finally, we review work across the chosen problem domain of intelligent transport systems, identifying the technological advances which we feel make the area a suitable near-future disruptive event. We review vehicle communication developments, before considering simulation of both individual entities, and a larger traffic collective. Assessing problems faced in this area, and the successful application of other computational techniques showing domain benefits, we draw the conclusion that this is a strong problem domain to ground our work in. However, we seek to retain a generalised approach to the problem rather than specific solutions for the transport domain, instead using transportation as a rich domain in order to explore the high level goals set out in the problem statements of Section 1.1.

## Chapter 3

# Analysis of the problem

An overview of the problem area is discussed in Chapter 1, and we now seek to identify generalised elements of the problem, expanding upon the content put forward earlier. In doing so, we aim to ensure that the work presented in this thesis holds relevance for a wider application area than just those demonstrated through the specific scenarios presented later in Chapter 5.

However, there is a need balance a generalised approach against demonstrable relevance, in order to address questions of how the theory can translate into practice. To address this, we define the problem in the context of intelligent transportation systems, and put forward scenarios which demonstrate the problem components being tackled against specific, domain-relevant examples.

Finally, we draw these discussions together to outline the system requirements for the software framework. These requirements identify what characteristics are required to support investigation into the problem statements put forward in Section 3.2 for a general set of problems, whilst allowing us to draw results from the the domain of intelligent vehicles, in order to assess the validity of our approach in addressing the problem statements.

### 3.1 Generalisation of problem type

Chapter 1 introduces the motivation and problems being addressed. We now revisit these in groupings of generalised problem topics which we consider as pillars in this work in relation to developing artificial socio-cognitive systems.

#### 3.1.1 Knowledge and understanding

Underpinning a significant part of this research effort is a set of problems which we group under the umbrella of ‘knowledge and understanding’. Relevant to both human and artificial systems, these terms can be interpreted to suggest differing levels of sophistication, for example to question how much human-like understanding an artificial system can have, before requiring sophisticated cognitive representation. Conversely, it could be argued that a simple artificial



system is demonstrating understanding of a situation if it is reacting appropriately to given stimuli.

In order to provide a framework for the topic of understanding, we draw on the work of Endsley [1995] in defining Situational Awareness (SA), comprising of an increasingly developed understanding that builds up from base level perceptions, to comprehension of the meaning of these perceptions, through to a high level projection element where the comprehension is extended to include a temporal element i.e. what may happen in the future and the impact of such events.

Having considered these as potential level descriptors for an artificial system’s awareness, we then need to address the question of what functionalities, or capabilities, are required to improve this understanding. First, we start with the low level perception element of SA, with the inference that unless the artificial entity has some sensor feed, or some method to be ‘told’ about environmental events, it cannot generate any perceptions. Therefore, if a sensor feed is to be established to an entity, questions arise as to how best to design this feed in order to generate SA. Do we attempt to feed all possible information and states to the entity, and develop reasoning and processing capability in the entity to make sense of this information? Certainly, that is an option, and an aspiration of areas such as the “Big Data” topic of research, but there remains the question of appropriateness of the information and quantity: does an entity need to know, or benefit from, that level of data? Is it possible that a more select level of data publication to the entity may result in the same, if not better, level of performance?

This question moves us into the comprehension element of SA; rather than feeding an entity a high volume of perception data, could some computation (i.e. data fusion, stream reasoning, as discussed in Section 2.1.2) be off-loaded elsewhere, and the entity fed with a richer (by which we mean higher SA, comprehension or projection related information) data source? Here we expect benefits in both reducing the computational effort and power required by the receiving entity, as well as a reduction in the load on the communications infrastructure in use. The processing location where this computation is offloaded to may also have access to a much larger data set than would have been transmitted to the target entity, and so may be able to add ‘comprehension’ to the perceptions which the receiving entity would have not been able to produce.

Considering the ‘projection’ element of SA, we see this as an addition to the requirements to support comprehension; even more resources are required to combine perceptions, comprehension, and time elements of the situation to draw conclusions about likely future states of the environment. In general terms, this expands the problem of how much to communicate even further, for how do entities determine what of their own state data to inform other entities about? Taking this argument further, how should an entity ‘package’ this information should it wish to share it? Even considering sharing simple perceptions, some definition of units used (e.g. kilometres per hour versus miles per hour) needs to be established, which becomes more challenging as we deal with more complex comprehension and projection elements where there is no obvious base unit.

As part of the generalised problem view, this brings us into the realm of the larger problem

area of ontology, with a requirement for both including definitions regarding what a piece of information being sent ‘is’, as well as a means for an entity to identify what elements of information it holds may be beneficial to others in the SA context (e.g. sharing an intention with another entity, so it can improve its projection element of future states).

### 3.1.2 Introduction of novel technologies

In this aspect of the problem area, we consider issues of introducing novel technology into a new area, and the situation where its impact, use case, and management are not always fully understood beforehand. Where a technology has been developed in tandem with consideration for how such integration into the world would look, requirements such as infrastructure may already be in place pre-deployment, ideally along with established legislation to govern the use of this new technology. However, when a new technology ‘explodes’ and is rushed into production, consideration of its use and integration with existing systems and procedures can lag behind.

One such example can be drawn from the introduction of vehicles on UK roads, where the legislation and integration with the existing means of transport did not keep up with the practicalities. In [Various \[1866\]](#) details of the ‘Locomotive Act 1865’ highlight how an existing approach to governing much larger steam locomotives was applied to smaller road vehicles, requiring one person to walk 60 yards ahead of the vehicle waving a red flag (despite a speed limit of 4mph and 2mph in towns). Eventually such legislation was revisited, removing the requirements for the flag waver, increasing the maximum speed, and moving closer towards the legislation which we find today.

More modern examples can be easily found, with questions being raised around ‘Google glasses’ (should one be able to capture everything as a video stream when its not obvious this is taking place?), and aircraft drones (both the management of airspace, and what forms acceptable use of such capability), where such cases also introduce ethical considerations and requirements. There are challenging problems posed by less dramatic examples too, for example the rise in ‘movies on demand’ content via the internet, with the issue of whether the service provider should carry the burden (whilst not receiving a direct financial reward), and whether there is even sufficient infrastructure to support such bandwidth demands, with conflict between Netflix (as a content provider) and Comcast (as an infrastructure provider), providing a current (as of 2014) example of this.

The penalties for mishandling the introduction of novel technologies can vary in magnitude and consequence. Perhaps the least severe can be seen when a technology is deployed without sufficient consideration for its use and benefits, for example the hype and deflation associated with virtual reality technology. In this case the negative outcome was consumer dissatisfaction and (somewhat) abandonment of the technology, though its quite possible the technology may be re-adopted at some point in the future.

More alarming penalties can be found arising from unforeseen events or complications around new technology. The theory of “black swan events” was put forward by [Taleb \[2007\]](#),

describing the occurrence of an event (a black swan being found) which disproves a belief set held prior to that observation (all swans are white). This concept is drawn upon here, as it is difficult to fairly judge whether an unexpected event relating to new technology could have been identified before its deployment, or was contrary to anything known about at the time. Historically we can see a number of cases where novel technology (or designs around that technology) ended badly, for example the Hindenburg disaster in 1937 bringing the end of hydrogen blimps, and the Air France 4590 accident in 2000 contributing to the demise of Concorde.

Moving away from mechanical-based examples, there can be unintended results arising from human interaction. Whilst some cases are fairly non-malign and easy to remedy (for example, Microsoft Office personal assistant ‘Clippy’ being dropped after receiving negative user feedback) some issues pose greater challenges to resolve. In a recent study, [Lichtenstein et al. \[2012\]](#) highlights that there may be growing trend of accidents where pedestrians have been wearing headphones. If this is due to distraction and not hearing any warning from a vehicle or train, then it is difficult to see how this unintentional consequence of headphone use could be easily remedied.

In software terms, extensive testing procedures may highlight issues, but drawing on the earlier black swan example, if a situation has not even been conceived of, or the circumstances that would give rise to such a case have been deemed unlikely to occur, then a test would not cover such an eventuality. Representing situations in software terms requires suitable modelling and simulation, and then introduces a number of benefits, allowing the potential system to be introduced to end users, stakeholders, legislators, so feedback to be captured. Interaction with existing devices and infrastructure can be modelled, and variables manipulated to explore permutations and ‘what-if’ cases. Furthermore, strategies such as monte-carlo simulation may be deployed in order to represent inputs and simulation configurations which have not been constructed by a human designer, in an attempt to alleviate the previously described black-swan phenomena. Improved human representation may add further unpredictability, but in the end we can mostly only deal with known-unknowns, while black-swans may remain unknown-unknowns.

### 3.1.3 Ability of intelligent systems

Another general problem when dealing with artificial socio-cognitive systems, is the ability of such systems to handle complex, problematic situations. Following the earlier discussion in Section 3.1.1 considering knowledge and understanding needs, and in general the problem statements set out in Section 1.1, there remains the problem of finding a suitable architecture which can deliver those requirements.

One problem around the choice and design of an artificial system is determining its suitability to perform a given task. Whilst with enough effort, it is often possible to force a particular design approach onto a problem, choosing appropriate technologies and abstractions of the problem can reduce effort and simplify solutions. For example, codifying intelligent agent behaviour by drawing on higher level AI programming languages rather than coding a new

solution from scratch.

There is also a need to address areas where intelligent systems may struggle compared to their human counterparts. Social interaction provides one such problem area, where the nuances of both human behaviour and cultural variation add complexity to the situation. Whilst two artificial systems may be able to resolve such situations between themselves (though that still presents challenges), when interacting with humans the uncertainty is increased significantly (e.g. due to vagueness and imprecision of communication). There are further problems of how such issues should be addressed, for example, through increased negotiation capabilities, an increase in cognition and reasoning ability, defining procedural rules for what to do in different situations, and so on. This is followed by the issue, of how such rules can be modified at run-time, for example if an artificial entity is operating in another country, an update to cover a different legal jurisdiction and cultural modifications may be required.

However, there are cases where artificial systems can out perform humans (e.g. with large volume data processing tasks and rapid decision making). This leads us to consider the general problem, of how can we overcome areas where artificial systems may struggle, whilst selecting a design which supports the observability of the artificial entities' decision making process for other BSF components and human debugging.

### 3.1.4 Constructing distributed systems

To support the process of investigation into artificial socio-cognitive systems, some experimental framework is required where different approaches can be prototyped and their impact assessed. Whilst it could be feasible to adopt a monolithic approach and combine all system components into one application to run on a single machine, there are a number of reasons to adopt a distributed approach. One such reason is that it provides a more explicit (and potentially observable) form of communication between components, as messages would need to travel over some infrastructure and so inherently it introduces an awareness of infrastructure requirements into the research. It also provides the option of adding computational resources, by running certain components on separate platforms.

However, consideration is required of how distributed components should interact and exchange information. As discussed in Section 2.2.3, there are a number of approaches to distributed simulation, and here we draw a comparison between two, in order to establish more general points. One approach is for simulation members to publish a predefined message type out to the network, and assume any interested parties will receive and be able to decode that message. Here we consider approaches such as the IEEE DIS protocol, but in general this can include any approach where some TCP/IP packet is sent, with  $n$  bytes representing variable X, followed by another set of bytes representing variable Y, and so on. This has the benefit of being a lightweight approach, but lacks sophistication both in knowledge representation as well as simulation management. As an alternative approach, the HLA model provides greater direction through its interface specification of how objects should interact, as well as including aspects of time management, but member have to follow certain guidelines e.g. joining

federations, publishing only predefined objects.

We consider the issue of simulation management (i.e. members just turning up and publishing data versus following predefined message specifications) as a broad problem when constructing distributed systems: the need to have a framework which offers flexibility yet where members are required to present their data in a certain fashion such that it can be understood by the rest of the community. This can be considered in the underlying message format as well, for example fixed structure, predetermined meaning (e.g. JSON) compared to unstructured, ontology determined meaning (e.g. RDF). The earlier problem statement of Section 3.1.1 put forward the need for an additional layer of description around information being exchanged, to improve understanding of what was being communicated, and we expand this further: that supplemental data is required for the additional reason of supporting components functioning in a distributed configuration. Such functionality adds support for both real-time analysis and experimentation, and post-simulation replay of events and offline analysis. Whilst aggregated or compound representations of data may have been presented in real-time, the ability to deconstruct and ‘reverse-engineer’ from low level data is required in order to construct new views and assessments (an analogy being retaining the original text of a PDF, whilst only the PDF may have been presented to users at the time).

## 3.2 Problem statements

Here we revisit the problem statements put forward in Section 1.1, sharpening the focus from the general problem discussions of Section 3.2 into specific areas to be addressed.

### 1. How to build awareness and understanding of a situation

For artificial entities to perform in a given environment, they need to understand what is going on in order to be able to function. This problem statement seeks to address what ‘depth’ of awareness and understanding is sufficient, whilst attempting to avoid over-simplification (resulting in unwieldy representations) and over-complexity (introducing heavy-weight solutions when simpler solutions may perform adequately). There is also consideration of an entity’s self awareness in terms of knowing what it knows, though there is at the same time a need to be pragmatic. Whilst these desires may seem to be in conflict, we seek to identify a middle ground where a suitable AI approach is able to perform within a given domain.

Challenges involve:

1. identifying a suitable approach, capable of meeting the awareness and understanding requirements, but which also
2. addresses the rest of the problem areas, and
3. demonstrates a suitable level of sophistication which could be applied to further areas (outside of the target domain area).

## **2. How can the reasoning process and information flow behind a decision be made observable**

Considering a wide variety of potential components which may interact with autonomous agents, we put forward a need to provide access to, and visibility of, the ‘inner’ workings of the agent deliberative process and belief set. Whilst agents within the agent framework may have such capability provided by the framework (e.g. via KQML/FIPA discussed earlier in Section 2.2.1), components such as governance frameworks may require such data to be made accessible (e.g. to understand why an action took place). This also introduces an additional benefit of providing a multiagent, distributed system debugging capability. The exposure of such internal processes, between the components of an intelligent system, that can be coupled with output from other system components, as a means to provide the system debugger with a view as to why certain events occurred, or did not occur.

However, this introduces a number of challenges: How much information should be, or is required to be, presented? What format allows the human to understand, without transforming the intelligent systems state such that it no longer represents the true underlying data? Is such information exchange feasible given infrastructure limitations? These provide just a few examples of the general challenges touched upon by this problem area, which require consideration in both the solution choice and experimentation.

Challenges involve:

1. Identifying an approach which supports observability both of internal component states, and exchanged messages, but which also
2. Can be handled both by system components (and in doing so, supports the rest of the problem areas) as well as a human debugging the system, and
3. Is able to operate at a variety of transfer rates.

## **3. Finding the right balance between quantity and appropriateness**

This is a broad problem area, relating to the earlier problem statement regarding building awareness and understanding, but also having ramifications for the requirements of supporting infrastructure and performance of system components. Whilst increasing technological capabilities have improved the size of data sets which can be processed, we consider a generic set of applications which could consist of a high quantity of entities, with varying computational power and communication bandwidth, where some limit on the amount of information exchange has to be established. Furthermore, even removing such limitations, there is a question of whether communicating less information, but of a richer content, may assist entities in better understanding the environment and others within it, which introduces the ‘appropriateness’ component of this problem statement.

We consider this problem from a Situational Awareness approach (as discussed earlier in Section 2.1); that information can be exchanged at a low level basic information (e.g. the perception element) through to high level complex data (e.g. the projection element). A set

of scenarios is required to assess varying communication approaches, to investigate the potential trade-off between quantity versus richness, and to inform the research of communication requirements between distributed intelligent systems.

Challenges involve:

1. Bringing together various components with different communication needs, in a framework which can
2. Support communication of different types of data at a variety of rates, whilst
3. Maintaining the same level of performance when communicating at different rates and levels of knowledge.

#### **4. How to resolve problematic situations**

For an intelligent system operating both with other artificial systems, and also interacting with humans, a difficulty arises in how to handle situations where the artificial entity has no solution, lacks suitable knowledge to comprehend what is occurring, or cannot recognise it is in a dead-lock situation where it lacks even the awareness that it cannot solve the problem. We use dead-lock in the conceptual sense of cases where an entity may be waiting on another entity (waiting on the original entity), or of a shared resource, rather than any specific computer science case. We consider the perspective that, whilst attempts could be made to encode information of all possible future events and corresponding actions to take, the permutations involved make it extremely difficult to cover all eventualities. This is compounded by the complication of interacting with humans, who may not behave in the expected fashion, resulting in a situation where the artificial entity is unsure what to do next.

Such problems can be found in plenty of real life situations, where humans enter into conflict and some external party is required to arbitrate or provide a new set of expected behaviour to the involved parties which carries some weight, such that the humans parties are likely to comply. This introduces further dimensions to this problem statement, of how such a process can be implemented within artificial systems, removing sufficient ambiguity and resulting in a resolution for the encountered problem.

Challenges involve:

1. Identifying a suitable approach capable of addressing such a requirement, where additional information can be contained
2. How dead-locks can be handled, where multiple entities may need to interact to achieve a solution, and
3. How to define such information in a generic fashion.

#### **5. How to balance individual pursuits versus the greater social welfare**

In giving an artificial entity autonomy of action selection to pursue a goal, there is some concern over what process that entity may take in order to achieve that goal. There is the primary

risk that the action choice may be to the detriment of others, and needs to be curbed in some fashion, but there is also the associated problem of whether by taking a larger view (e.g. society-centric rather than individual) that the greater population could benefit. Such modifications to individual behaviour may prevent the individual from achieving an optimum result, but could improve result for the wider collective. We recall the earlier discussion from Section 2.2.2 regarding the “tragedy of the commons” problem, and highlight this problem statement as seeking to address this issue.

This introduces the problem of how to achieve a coordination activity across a number of distributed autonomous entities. There is the question of what structure would allow such enforcement, what and where the decision-making process behind such outcomes should be, and how to provide guidance to an entity such that the required behaviour is adopted, without constraining autonomous behaviour for other objectives.

Challenges involve:

1. How to provide guidance to artificial entities based on events and states within an environment, whilst
2. Maintaining their autonomy when implementing such guidance, and
3. How entities return to prior behaviour once such guidance has been implemented.

## **6. How to demonstrate the application of the principles we have identified, whilst retaining a general solution**

There is a challenge involved in establishing the effectiveness of the application of high level concepts and approaches, with a need to transition from the theoretical to the more tangible, grounding the theory in a real domain application to raise questions and test the assertions made in the theoretical approach. This addresses questions of whether a chosen approach can work in practice, and provides results in the target domain which can be reviewed by that domain community to assess plausibility.

Such a goal creates a conflict, as the general applicability of a theoretical approach needs to be maintained, whilst at the same time producing credible results in a specific domain to prove the approach. This problem statement exists to ensure focus on this task throughout the work, that developed solutions and findings should be relevant cross-domain, whilst the work should be grounded deep enough in a targeted domain such that meaningful, credible results can be generated.

Challenges involve:

1. To construct sufficiently complex scenarios in the domain of intelligent vehicles where meaningful results can be generated, whilst
2. Ensuring the developed framework can be applied to alternative domains, and
3. Identifying the wider implications of the findings than just within intelligent vehicles.



## 7. How distributed systems can support a variety of component types for a generalised set of applications

The previous problem statements have touched upon requirements such as knowledge representation, communication approaches, and the need to demonstrate theory in practice. Whilst the components to make up such capability could be co-located on a single computer, this seems an unlikely configuration given the availability of external knowledge sources, additional computational power, and that artificial entities may be geographically dispersed, such as on board the device they are controlling.

This introduces a set of problems related to distributed systems, with a need to consider how components can run from various (computer hardware) locations, how they exchange information, and how this process of communication and action can be reviewed by a human observer to investigate what happened. Whilst the topic of ‘distributed debugging’ was introduced in problem statement two from an observability perspective, here there are problems in respect of how this data can be analysed, replayed, and combined with other data points, in order to assess the performance of a framework against a particular problem.

Challenges involve:

1. Identifying a suitable distributed systems engineering approach, which
2. Demonstrating the ability to integrate disparate component types, whilst
3. Retaining application to a broad set of problem types.

### 3.3 Defining problem in a specific domain

In order to transition these generalised problem statements into a specific set of research questions, a problem domain was targeted, with a number of desired characteristics. The first was that of complexity, where the problem domain needs to have sufficient complexity such that the approach of this work can be demonstrated meaningfully and generate interesting results (i.e. some expected, some unexpected, some of potential benefit to the domain). However, the complexity needs to be variable; for some experimentation it may be desirable to reduce the level of background ‘noise’ and focus on a small number of factors, whereas for other scenarios it may be desirable to introduce a large amount of noise and complexity for the intelligent system to handle (i.e. to stretch processing capability, bandwidth, reasoning against large datasets).

The chosen problem domain also needs to be suitable for the socio-cognitive components of the problem statements, and so include areas which involve collaborative activities where reasoning is required about the actions of other entities. In order to demonstrate the general application of the research, and to maintain a challenging level of complexity, it is desirable that the problem domain involves interaction with humans (or a simulated human representation), rather than being limited to interaction with only artificial systems.

Whilst the problem statements discussed earlier in Section 3.2 do not explicitly target this requirement, selecting a problem domain where there is new and potentially disruptive technol-

ogy on the horizon introduces a number of benefits. Such technologies may offer the opportunity to integrate with other software components, for example to extract sensor information, or some control over the functionality of that new device (e.g. cleaning robots with APIs, kitchen appliances with IP addresses, home automation, and so on). This provides a means by which the framework being developed could be coupled to such devices, giving the intelligent agent component the ability to interact with the new technology.

Such interaction with a new technology brings additional benefits as it offers the ability to explore what-if scenarios. Discussed earlier in Section 3.1.2 was the example of legislation not keeping pace with technology developments, which can be seen at present with debate around the moral and legislative aspects of autonomous devices. Selecting a domain in which the use-case and interaction with existing (perhaps human) users is not fully understood allows us to demonstrate the capability of prototyping scenarios to explore the ‘what-ifs’. This can help in clarifying a spectrum of issues, from how much communication bandwidth is required based on a given message exchange approach to what happens for a given set of capabilities if devices are allowed full autonomy.

The area of intelligent transportation systems is identified as a complex application area, containing social interaction with humans, where new technology is being introduced along with elements of automation in self-driving cars and vehicle convoys. There are a number of relevant developments in the area of autonomous vehicles (see Section 2.3.1) along with communication between vehicles becoming possible (see Section 2.3.2). Based on these developments, we consider potential future scenarios resulting from this technology becoming more widespread, introducing a variety of socio-cognitive problems.

We seek to investigate these problem statements through the construction of suitable scenarios exploring self-driving vehicles in a mixed human-artificial entity environment. The first set of scenarios is based on vehicle convoys, where a lead vehicle is heading to a certain destination and knows the route to that location, and a number of convoy members wish to go to the same place but either do not have the route details, or just wish to follow the convoy leader for some distance. In this area, focus is on communication between vehicles and how they use this information to plan and react accordingly. Such focus on communication provides the opportunity to consider the problem area of observability of information, and the varying communication strategies based on the Situational Awareness concepts. We propose a number of scenarios to explore specific aspects of SA in the domain of intelligent vehicles. In Section 5.2 we explore the performance of a vehicle convoy when exchanging low level SA percepts of position updates at high versus low frequency, followed by Section 5.3 which considers exchanging a higher SA level of route information being exchanged and the corresponding effect on convoy performance. In Section 5.4 we consider the high level projection element of SA, exploring how an awareness of future traffic light states can allow a vehicle to adapt its speed in order to arrive at the traffic lights when they are green. We adopt these experiments in order to highlight how consideration of SA for intelligent vehicles in the traffic domain can lead to both individual and collective benefits, without increasing throughput demands on the communication framework.

A further set of scenarios considers issues relating to the autonomy of these intelligent

vehicles, seeking to investigate how vehicles can receive additional guidance in situations where they are unclear what to do, or where their own autonomy may lead them to take actions to the detriment of wider vehicle population, and thus need some corrective measure.

Details of specific scenarios to investigate these areas are presented later in Chapter 5, but the general requirements of such work are now presented.

## 3.4 Investigative system requirements

In this section, we put forward a set of requirements with the intention of creating a framework to support experimentation based in the problem domain outlined in Section 3.3 against the problem statements put forward in Section 3.2. These requirements also consider the generalised problem discussions of Section 3.1, and we seek to construct a system capable of supporting research in the domain of intelligent vehicles, but designed in such a way that the components brought together remain abstract and applicable to other problem areas.

From a systems engineering perspective, the discussion in this section does not closely adhere to either functional or non-functional requirement specifications, though of the two the details sit closer to non-functional requirements. However, the requirements are not developed to the extent of being fully stated, for two key reasons. Firstly, it is difficult to know ahead of time what the exact requirements would be, for example, we can assume reliable message exchange is a requirement, but how reliable? It may be desirable to observe how system components handle such loss, rather than over-specify a lossless system. Secondly, the full statement of requirements may vary depending on the application domain, for example a shift in domain may change a requirement from supporting hundreds of entities to hundreds of thousands. As we wish to retain a general applicability to multiple domains, we instead seek to identify more generalised requirements for the *type* of functionality, rather than its exact specification.

### 3.4.1 Information exchange

The approach to information exchange is a key design consideration which will have an impact on all aspects of experimentation, as well as the ability of the constructed system to address the problem statements set out earlier in Section 3.2. Within the area of information exchange, we put forward a number of characteristics which the developed system will need to support.

#### Distributed

The topic of distributed systems was introduced earlier in Section 3.1.4, with the objective of avoiding large single platform implementations, and instead adopting a distributed approach, where code and components of the framework are able to work across multiple computer platforms connected via a network.

The requirement is that components of the framework are able to send and receive data, without being closely coupled to another component, i.e. without any special processing requirements, dedicated links between specific modules, or mismatch of data (e.g. frequency, message

length) causing any components to fail. Furthermore, it should be transparent to members of the framework if a component is replaced, so long as the same message specifications are used.

There should also be no specific dependency on the time of message receipt (for the sake of loose coupling), rather this should be included as a component of the message itself and handled accordingly, as the framework may be used for real-time work, but also slower than or faster than realtime experimentation. The framework may also be used for after event analysis, replaying historical logged data, but with framework members responding in the same way as if this were live data. Related to this, the same outcome(s) from experimental scenarios should arise both when repeating that experiment with live data, as when using logged or simulated data. This requirements can be summed up as: time-independent, replayable, and repeatable.

## **Observable**

One requirement directly linked to the problem statements of Section 3.2 is the need for the data (which includes messages but also internal states of components) to be observable, in the sense of being able to extract it, but also that it is meaningful when taken as a single data piece. The extraction requirement will be supported by the distributed design approach, but when extracted the information still needs to be intelligible, as the ability to look at a single message exchange and understand what was being communicated (and where it originated from) is extremely useful.

There are a variety of approaches which could be adopted to provide an explanation of transmitted data, for example adopting some standard where packets are of a predefined type and format (e.g. the DIS approach) and can thus be decoded back to their meaning. However, such pre-definition of data formats pushes the implementation into specific problem areas (e.g. DIS may suit military representation, but be less suited to other areas). Instead, we require the data to be annotated in such a way so as to become self-contained, both in definition of what it is, as well as the time element of when it was sent.

We consider such explanation and mark-up of the data as semantic annotation, adding a descriptive layer around potentially unstructured data, allowing recipients of this information to reason about the message being received e.g. who (or what) it was from, what the subject of the data is, and so on.

Finally, considering the requirement of data extraction in order for the information to be observable, there may be cases where framework components would not naturally publish their own internal state data, but that this may be of interest to others. For example, this could be the refresh rate of a graphics engine, an internal belief state, or the current time step of a simulation component. This leads to the requirement, that where possible open source components should be adopted, so that if additional information is required, the code can be modified and this data published to the framework.

## Interchangeable

Following on from the observability requirement, and touched upon with the distributed requirement, we revisit and state explicitly that information sources (i.e. providers of data to the framework) should be interchangeable. For example, it should be transparent to a message recipient whether a received message originates from a simulated or real device. This is to allow components to be replaced with other alternative components, so long as they comply to the same message specification. For areas of this work where no established message type exists, this will require a specification of what messages should be and at what level of abstraction (e.g. units of measurement, as well as what is suitable to be measured).

This extends to the time element of messages, and framework members should respond to a live component providing some data stream (e.g. vehicle positions) in the same way as to another module replaying this data from a logged source.

## Data rates

Another aspect of information exchange is the issue of data exchange rates. This proves to be a challenging requirement to specify before experimentation, as numerous factors are unknown. For example, one aspect of this is driven by hardware and software limitations, in order to avoid message delay, saturation of physical networks, etc.. Related to this, is the appropriateness of exchange rates: whilst a 3D rendering engine would be expected to run somewhere above 15 frames per second (and so potentially require 15 updates a second from some publishing component), an intelligent layer may become saturated at such a rate, and be better served by 1 update a second.

This introduces the requirement of ‘impedance matching’, where consideration is required to match data flow rates of component with their operational cycle rate in order to ensure reliability (i.e. appropriate and timely responses). Too far away from this notional optimal value, and there is either data deluge, or insufficient data. Both of these situations increase the likelihood of incorrect action selection, as the processed data at a given decision point may be stale. Therefore, the requirement becomes that sources provide various ‘channels’ of data streams in some fashion, with the capability for recipients to choose which they receive data from.

### 3.4.2 Intelligent control

Having selected the problem domain of intelligent vehicles, there follows the requirement to provide some intelligent, autonomous behaviour and control to these vehicles. There are two initial requirements, firstly that vehicles react to events and are able to achieve goals based on their understanding of the environment, i.e. deliberative behaviour. Rather than attempting to specify such control at a low level, a higher level interface is desirable in order to retain a degree of abstraction (i.e. while the problem domain is focussed on vehicles, the solution should remain applicable to other areas). Error and failure handling is desirable, since intelligent agents will be operating in a complex environment where they may have insufficient knowledge to handle

every possible event, the adopted framework should provide some coping mechanism for the unexpected.

We put forward the requirement that this should be a multi agent platform, in order to allow multiple intelligent agents to pursue their own individual goals (and interact with other agents in doing so), and this also fits the model of integrating with an external environment through the use of appropriate sensors to draw percepts from the environments, and actuators to bring about some action in the environment.

From the requirements put forward in Section 3.4.1 regarding information exchange, the intelligent component needs to be capable of integrating into the overall framework, making its internally held data available and also treating the data provided by the framework as its environment, where intelligent agents can effect that environment through actions transmitted out to the framework. Related to this integration, is the requirement for this component to operate in realtime, capable of processing new environment data and appropriate intelligent agent responses within a suitable timeframe.

### 3.4.3 Governance

A number of problem statements set out in Section 3.2 suggest there is a requirement to establish some external body, which is capable of providing both resolution and guidance to the intelligent agent layer. We consider this a requirement to introduce a governance structure, where events occurring can either be detected by the governance module or referred to it directly, and result in the issuance of some instruction or clarification.

We require this capability to be non domain-specific, but instead driven by a set of self-contained declarations and configuration, triggered by the receipt of data from components in the framework and providing a response back to the framework within an appropriate time duration.

The guidance provided by the governance component should not be considered mandatory, as the receiving intelligent entity may have an appropriate reason to not follow the instruction (e.g. an event may have just occurred which would make it dangerous to do so), and there is a requirement to handle of such violations. Similarly, there may be a requirement to indicate some time component for issued instructions (e.g. perform action for this length of time, ignore if not implemented by this time).

### 3.4.4 Problem domain representation

As this research is grounded in the problem domain of intelligent vehicles, there is a requirement to represent this domain adequately for experimental purposes, where the various scenarios can be constructed and used to provide a simulated environment to the rest of the framework components.

A simulated domain representation is required, such that unsafe scenarios (e.g. vehicle crashes) can be investigated, and that untested intelligent agent control can be assessed without risk of damage. Furthermore, the simulated approach is required to provide consistent,

repeatable output, such that the behaviour of the other framework components can be assessed without dealing with random events and real world sensor noise. Although simulation has been selected in order to simplify the problem domain representation, it still needs to be of suitable credibility such that the intelligent agents are challenged and receive sufficient environment information for the scenario under investigation.

Whilst for the research in this work, the requirement is for this to be a simulated capability, from the interoperability requirement of Section 3.4.1, it should be possible to replace this with a real data feed without significant effort and impact on existing components.

The simulation package also needs to be integrated into the overall framework, and so there is a requirement that it can be customised sufficiently to send vehicle and environmental data to the framework, and also respond appropriately to control input from intelligent agents seeking to affect vehicles they are controlling. This poses quite a significant operational requirement, meaning we require a simulation package which can provide some representation of autonomous background traffic, but also turn off such internal behaviours and let an external platform control these for specific vehicles (e.g. speed control, lane change decisions, collision avoidance).

### 3.4.5 Review of results

As the research activity is focussed on experimentation, there needs to be some method(s) of analysing the results. The difficulty is that ahead of time it might not be possible to know the entire set of required measurements. For this reason, the ability to log data exchange within a simulation is desirable, in order to improve existing metrics and develop more, against the same experimental data. This is coupled with the requirement to present such data in a variety of formats, which may not be known ahead of runtime.

#### Logging and replaying

There is a requirement to log data from the framework, along with the ability to replay it. Aspects of this have been introduced in the requirements of information exchange put forward in Section 3.4.1, where the information itself should be in a format such that it can be replayed, and this requirement considers the need for a software component to perform that task.

There are also a number of requirements regarding the ability to replaying data. Regarding the problem statement of a reasoning process being observable, the replay capability may need to provide timeline changing capability (e.g. stop, rewind, resume) to allow the human observer to review scenario events a number of times.

The logging capability needs to log the data in such a format that supports the replay component, but also that allows offline analysis of the data. For example, whilst the replay component may step through each time segment of logged data and send out to the framework members, offline analysis may build queries regarding the number of messages of type X, the frequency that entity N published message Y, and so on.

### 3D

Although not explicitly required by any of the problem statements, the ability to visualise observed data made available through the requirements of Section 3.4.1 in 3D space is included as a requirement, in order to assist a human debugging the system to understand what is occurring. For the targeted problem domain of intelligent vehicles, this capability would allow a 3D representation of the vehicles and their environment, which even at that basic level allows an assessment of whether the vehicles are behaving as expected (i.e. travelling in a straight line to a chosen destination).

However, we expand this requirement, that information from other data sources should be represented in the 3D view. Here we wish to make use of the data extraction process in order to gain visibility of the inner workings of other framework members. The key components we wish to view in the 3D representation are drawn from the intelligent agent platform, and from the governance framework. Therefore, the requirement is the ability to blend disparate data types in order to present a fuller view of what is occurring, to assist in understanding the environment and also debugging as needed.

Finally, this view capability should also provide a basic GUI to the human operator, with controls of time replay linked to the requirement of logging and replaying. We intend this 3D view to become a central focus in understanding events in the environment, and when using logged data it would be beneficial to be able to pause, rewind, resume, in order to replay and understand the causal chain of events.

### Analytics

Having set out the requirements for logging, replaying, and 3D representation, the final requirement for reviewing data is an analytical capability. As part of the review capability there is a need for runtime analysis tools, as well as offline analysis capability. The runtime analysis tools are required to assess whether a simulation is performing as expected, and capture particular situations as they occur, presenting these to the human observer analysing the system behaviour. This requirement is for a 2D graph capability, to provide both infrastructure metrics as well as metrics drawn from the target domain of intelligent vehicles. This capability should be able to save its current data set or view (i.e. to CSV file, or to a BMP of the current graph) for further analysis or inclusion in the presentation of results.

For offline analysis, the requirement is to be able to query logged data to draw out specific areas of interest, (e.g. how many instances of X occurred during the experiment).

## 3.5 Summary

In this chapter, a number of general problems relevant to socio-cognitive intelligent systems were discussed, of knowledge and understanding, human-machine interaction, the introduction novel technologies, the ability of intelligent systems, and constructing distributed systems. These



general problems were the distilled into specific problem statements which we will focus on throughout this work:

- How to build awareness and understanding of a situation,
- How can the reasoning process and information flow behind a decision be made observable,
- Finding the right balance between quantity and appropriateness,
- How to resolve problematic situations,
- How to balance individual pursuits versus the greater social welfare,
- How to demonstrate the application of the principles we have identified, whilst retaining a general solution, and
- How distributed systems can support a variety of component types for a generalised set of applications

To ground the investigation of these problem statements in real world problems, we propose the use of the intelligent vehicle domain, where we will create suitable scenarios to explore these statements, whilst retaining a generalised approach both in the developed architecture (where it remains capable of operating in other problem domains) and the findings (where we assess domain specific performance, but relate this to higher level generalised findings against the problem statements).

To carry out this investigation, we put forward a number of system requirements, from which we develop a generic framework which is detailed in Chapter 4, before implementing the scenarios outlined in Chapter 5. These requirements consider information exchange, intelligent control, governance, representation of the chosen problem domain and the ability to review results from the scenario experiments put forward Chapter 5.

## Chapter 4

# Developing an architecture and software solution

In order to support investigation into the problem areas put forward in Chapter 3, a suitable software architecture needs to be assembled in order to address both the generic high level problem tasks, but also to be applied to the specific domain of intelligent transportation systems.

There are various solutions existing to individual requirements of this research (e.g. intelligent agent frameworks, messaging servers), but no off the shelf solution was found which incorporated all these into an open framework suitable for our aims. In Section 2.2.1 a variety of intelligent systems were reviewed along with organisational and governance structures in Section 2.2.2. At the same time, communication approaches were reviewed in Section 2.2.3, and we now draw on these, in conjunction with the problem domain specific material of Section 2.3 to bring them together to construct a system capable of and suitable for exploring the problem statements put forward in Section 3.2.

Requirements were put forward in Section 3.4 that the developed system must provide information exchange which is i) distributed, ii) observable, iii) interchangeable, and iv) able to support a range of data rates. Further requirements include the need for intelligent control, a governance framework, problem domain representation, and to provide mechanisms with which to review the results.

In this chapter, we discuss the design of an architecture and its implementation to meet these requirements; in Section 4.1 we provide technical details of the messaging framework, intelligence layer, representation of the problem domain, the adopted institutional framework, along with general utilities to log and replay data, generate graphs and a 3D review tool. We present details of the evolution of the software architecture, as findings regarding the design of distributed systems were found during this process which have relevance to the problem statements of Section 3.4. We then present specific excerpts of the intelligent agent plans in Section 4.1.2, as these form the generic agent capability used in the rest of the work, whereas institution specific listings are provided per scenario later in Chapter 5. Finally, we present

some framework specific testing in Section 4.2 in order to establish framework performance, in respect of the general problem areas of communication and appropriate quantities for message exchange.

## 4.1 Components and interfacing

The requirements set out in Section 3.4 provide guidance in what components are required to investigate the problem statements set out in Section 3.2, and the type of functionality these components need to provide. However, there is (intentionally) not sufficient detail in these requirements to build a system from the outset. It is not known ahead of time what findings will arise from the experimentation, and if we over specify the design and the implementation becomes too rigid, it may preclude taking the research in certain directions later. Furthermore, there are specific problem statements in Section 3.4 concerned with retaining a generalised solution, based on a distributed systems approach, where we believe retaining a more open-ended implementation solution will be of benefit.

With this in mind, we take an evolutionary approach, where whilst we have established what component types are needed (in Section 3.2), the development is driven by domain and technical requirements. This section presents a technical overview of this development process, largely focussing on details of the final implementation. The topic of entity representation is discussed in more depth, and the evolution of alternative approaches taken assists in understanding the problem statement concerned with developing a generalised application.

We first present what are considered core components to support the creation and simulation of the scenarios detailed later in Chapter 5. In Section 4.1.1 we discuss the messaging framework which allows modules to exchange information, and we provide details of the message exchange format. In Section 4.1.2 we discuss the intelligence layer, and how the coupling between a simulated environment and the agent layer is implemented. In Section 4.1.3 details of the environment representation are presented along with alternative entity types and domains which have been considered in order to address the wider application areas. In Section 4.1.4 the institution concept and framework are presented, with details of standalone institution use, integration within the BSF framework, and the use of multiple interacting institutions. In Section 4.1.5 we discuss the Area Of Interest module as used to extract data of certain types from specific areas in order to provide a data fusion capability. With these components, the scenarios presented in this work can be performed, however, there is a requirement for a review and analysis capability. Details of a number of utility components are presented in Section 4.1.6 which support testing of the framework, logging and replaying of messages, and a monitoring capability. Finally, in Section 4.1.7 we present a 3D viewer which has been developed primarily to support the review of scenarios performed by the system designer (e.g. to view vehicles as they move along a route).

#### 4.1.1 Messaging framework

The motivation for exchanging semantically rich information was introduced in Section 2.1, with more specific requirements put forward in Section 3.4.1, and it is with this in mind that we introduce the messaging component of the adopted solution.

For the exchange of message types such as that shown earlier in Listing 2.1 and following in Listing 4.1 (i.e. in XML format) two approaches are considered, that of message queues, and web services. In the category of web services, we consider technologies such as SOAP and RESTful services, which allow a client-server like exchange of information. Whilst such an approach could play a useful role in supporting an entity in reasoning about its SA (e.g. to retrieve additional definitions, to check other facts, etc) and also assist in the problem area of ‘dealing with the unknown’ (e.g. referring some handled event or conflict to another entity for resolution), in terms of the communication discussion held in Section 2.1.2 there may be some drawbacks. The notion of knowing when to seek information does not entirely fit with this requirement, becoming more of a ‘data-pull’ than the ‘data-push’ outlined in the earlier discussions. By comparison, message queues operate with clients adding their message to a queue, which can be forwarded to other clients by the message queue server. This allows clients to express an interest in a certain message queue, and then receive new messages from that queue whenever they are posted, an approach taken further in the publish-subscribe methodology. This forms part of the XMPP (XMPP Standards Foundation [no date]), defined in the XEP-0060<sup>1</sup> definition where in summary ‘nodes’ are created on which information is published, and subscribers of that node receive the new messages. This forms a useful capability with respect to the problem statements set out in Section 3.2 regarding appropriate communication, as clients can choose which nodes they wish to subscribe to (e.g. a node containing high frequency low SA like information, or a node for low frequency but high SA, depending on the requirements and processing capability of the entity). Other entities do not have to consider explicitly the requirements of other clients with this approach, they simply send what information they have agreed to publish. This is where semantic annotation adds further strength as well, empowering the recipient to manipulate received data to a format it can do something with.

The Bath-Sensor-Framework (BSF) is an open framework to support a pragmatically simple approach to message exchange between components, and was adopted and contributed to during the course of this research, as it offered the functionality, flexibility and access to other components needed (for example, work had already taken place to integrate institutional models into the framework). Based upon the Extensible Messaging and Presence Protocol (XMPP Standards Foundation [no date]) protocol, it adopts a publish-subscribe methodology representing publishers as ‘sensors’ and subscribers as ‘sensor subscribers’. Messages are defined via the Resource Description Framework (RDF) (discussed in Section 2.2.3) which provides semantic annotation for the messages being exchanged. The BSF framework was originally conceived of for use in the context of smart homes, with sensors within the home publishing readings (e.g. temperature, light level, gas levels, window and door states) combined with control of potential

---

<sup>1</sup><http://www.xmpp.org/extensions/xep-0060.html>, accessed 25th September 2014

Listing 4.1: An RDF annotation generated by the BSF test package

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://bath.edu/sensors/predicates#"
  xmlns:j.1="http://127.0.0.1/sensors/types#" >
  <rdf:Description rdf:about="http://bath.edu/sensors/readings/
    D109F9806459B21D5CABE7A17F925FFA">
    <j.1:spatial rdf:datatype="http://www.w3.org/2001/XMLSchema#string">0.0, 0.0,
      0.0,20</j.1:spatial>
    <j.0:locatedAt rdf:resource="http://127.0.0.1/vehicleSensors"/>
    <rdf:type rdf:nodeID="A0"/>
    <j.0:takenAt rdf:datatype="http://www.w3.org/2001/XMLSchema#long
      ">1403193764198</j.0:takenAt>
    <j.0:isDataReading rdf:resource="http://127.0.0.1/vehicles/rdfTest"/>
  </rdf:Description>
</rdf:RDF>
```

actuators (e.g. radiator valves, window closers), and it retains applicability to other domains through the adoption of the publish-subscribe model (with new domains simply having to publish and subscribe to appropriate data channels). In the work presented in this thesis, we use the BSF as the message exchange mechanism in the context of intelligent vehicles, and in the scenarios presented in Chapter 5 explore alternative communication approaches regarding situational awareness. Throughout this work, we use the RDF mechanism as a means to enable semantic annotation of exchanged messages, with a shared ontology between the BSF components. Whilst the RDF triple is used as a means to add definition to exchanged messages, there are a number of limitations to the current use. The first limitation we see is that there is no additional definition or reference source available to reason further about what the received message is; the subject and predicate content of a received message have to be used ‘as-is’ (i.e. if an entity does not recognise the meaning of the message, no further assistance is currently available). The second limitation follows on from the first, that BSF members are required to have a common ontological understanding of these labels ahead of runtime. Whilst the adopted architecture may offer the capability to address such limitations, within the current work we assume (and implement based on this assumption) that such ontological compatibility between senders and receivers of BSF messages exists.

Listing 4.1 shows an RDF annotation generated by one of the BSF test tools (these tools are described later in Section 4.1.6), which we use here to see a typical example of information being sent by a BSF sensor, and the additional semantic information which is contained within a single message. Firstly, it has a unique ‘Description’ value, followed by a defined ‘spatial’ type (here containing the measurement in terms of position data). There is then a reading as to where the sensor is located, a time stamp (as an epoch value e.g. the value converts to Thu, 19 Jun 2014 16:02:44 GMT), and finally the source which generated this reading.

We have two relevant motivations from the problem statements of Section 3.2: that we

wish to support the observability of information flow (which the semantic annotation shown in Listing 4.1 assists in), and the need to retain a generalised, distributed solution. For this requirement, the BSF framework offers benefits in both the encoding of the message, and the code base around the publishing and subscribing of messages. For a given XMPP messaging server, the relevant standard (XEP-0060<sup>2</sup>) ensures that a similar service provision will be made regardless of the server software choice, i.e. there will be an ability to create nodes, to read and publish from these nodes, and to broadcast events when there is new data to be read. Therefore, so long as BSF publishers and subscribers conform to the appropriate standard (or make use of appropriate libraries), then components can be written in different languages, and still operate within the framework with other components. Furthermore, so long as the XMPP server is compliant with appropriate standards, this central component can be replaced with another implementation, which is explored further in Section 4.2. From a BSF Java component perspective, message exchange is based on the use of `DataReading` objects, with appropriate functions to encode readings into the RDF format, which is then published to the desired node resident on the XMPP server.

The composability of the system is taken further, where the message encoding itself can be changed to an alternative format. In this work we are proponents of the RDF approach, as a means to annotate data with additional semantic description, suitable for reasoning about by the recipient using established libraries. However, we also consider the problem statement of Section 3.2 relating to the balance of quantity and appropriateness of data. In the SA context, there may be some components that would be better served with a high volume of low level perception level data, and it may be that the RDF approach is less suitable in that situation. For this reason, we note the capability of the BSF framework to use an alternative to RDF, which has been implemented in other work (Lee et al. [2013], Lee et al. [2012]), based on the JavaScript Object Notation (JSON) format. Compared to the RDF approach, JSON is based on attribute-value pairs and so some of the additional annotation expressed in the RDF message is lost, but in some cases (i.e. simple, high volume data exchange) this may not be an issue. Furthermore, for different message volumes, such additional data may become an unnecessary overhead, and we discuss related performance differences between RDF and JSON approaches later in Section 4.2. The use of the BSF framework presented in Lee et al. [2013] also relates to the problem statements of Section 3.2 regarding the generality of the framework, demonstrating its use in reasoning about polite virtual agents based in the Second Life framework.

Effort is also made to support multiple device types in order to support potential new domains, through the use of alternative programming language implementations. Whilst the work presented in this thesis is primarily implemented in Java, other work (Lee et al. [2013], Lee et al. [2012]) has developed an implementation in the C# language. The Java approach has also been successfully used on Android and Raspberry Pi devices, demonstrating the use of the framework in mobile and low power contexts, and so broadening the potential domains where this framework could be taken.

Performance issues have also been investigated elsewhere, considered in Xiaoping Che [2014],

---

<sup>2</sup><http://www.xmpp.org/extensions/xep-0060.html>, accessed 25th September 2014

with work to formalise the problem across areas of XMPP, using a load testing tool (Tsong) and finding a performance drop off as message volume increases. However, it becomes difficult to generalise such limitations, as network infrastructure, software design, and messaging server implementation all introduce significant variation. Work has been performed by [Laine and Säilä \[2012\]](#) using the ‘ejabberd’ XMPP messaging server to investigate the suitability of XMPP for web based applications, and finds message exchange rates related to message size, with approximately 2000 msg per sec for 100 byte size message. The relationship of message size to data rate is an interesting topic given the SA context, and again raises the need to be mindful of what is communicated and the impact this has on the rate of message throughput.

We highlight the various implementation options available to show that simulation scenarios, agent behaviours, etc, retain a level of abstraction from any specific nuances of middleware implementations and that different components can use different representations to suit their needs.

#### 4.1.2 Intelligence layer

Similarly to the messaging framework, there are a variety of agent platforms which could be adopted, as discussed earlier in Section 2.2.1. One of the strengths of the adopted BSF framework is the ability to easily exchange one component for another, and this applies to the intelligent agent layer as well. We consider this in terms of Wooldridge’s situated agent diagram ([Wooldridge \[1999\]](#)), as reproduced in Figure 4-1, where in BSF terms, messages are received via subscriptions to various sensor feeds, and the received data represents the environment for the agents to act in, with their beliefs being updated from this environment, and their actions being published back to the framework as their sensor output, to be acted upon as appropriate by other system components.

In this section, we present the integration between the agent platform and the BSF framework developed to support this and meet the requirements set out in Section 3.4.2, and provide an overview of developed agents and their capabilities.

#### Agent platform and BSF Integration

Developing from Wooldridge’s situated agent concept shown in Figure 4-1 ([Wooldridge \[1999\]](#)), we recall from the discussion of Section 3.3 that the environment will be based in the domain of intelligent vehicles, details of which follow later in Section 4.1.3. From the problem statements of Section 3.2 regarding developing a distributed approach, as well as observability of communication, we intend to transmit the action and sensor input over the BSF framework as RDF messages (as introduced in the previous Section 4.1.1).

In the related research discussions of Section 2.2.1 a number of agent platforms were discussed, with focus on a number of research efforts which had demonstrated the Belief-Desire-Intention (BDI) model controlling various entity types (e.g. UAV, avatars, virtual footballers). Considering the problem statements of Section 3.2, and requirements put forward in Section 3.4.2, we are drawn to the BDI model for a number of reasons, primarily that its folk-

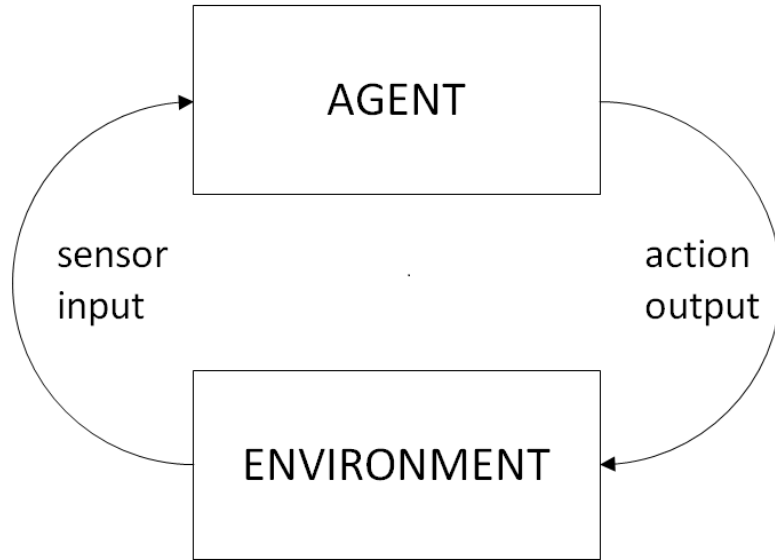


Figure 4-1: Reproduction of Wooldridge’s situated agents

psychology roots assist us in the observability of its reasoning process where human-like terms can be easily exported and made available to other BSF components. From the BDI models discussed in Section 2.2.1, the Jason (Bordini et al. [2007]) agent platform is our chosen platform to use in this work. The architecture of Jason lends itself to the situated agent model, with BDI agents having their beliefs updated from sensors based in the environment, which trigger plans and events in the agent, which can then select appropriate actions to bring about its desired goal.

In order to achieve this behaviour, we need to extend the Jason agent framework to interact with the BSF framework appropriately, so that the agent architecture can consume messages which represent events and activity in the environment, update the appropriate agents, which can then perform their reasoning and deliberative planning. Based on this, the agents issue actions where appropriate, which are then transmitted as BSF messages such that the environment can respond. To support such work, Jason provides two classes which can be extended as required, an `Environment` class used to represent an environment, and an `AgArch` agent architecture class which can support customised actions, and we draw on both classes to provide the required functionality.

To implement the environment of the situated agents and to connect to a source of events, the `Environment` class within Jason has been extended and implemented as `XMPPWorld` class, which subscribes to BSF nodes publishing relevant information (for the chosen problem domain, this would typically be vehicle information). This performs an element of pre-processing, in extracting appropriate information from the RDF message and translating this to percept updates suitable for the agents to consume. This includes appropriate forwarding of the percept



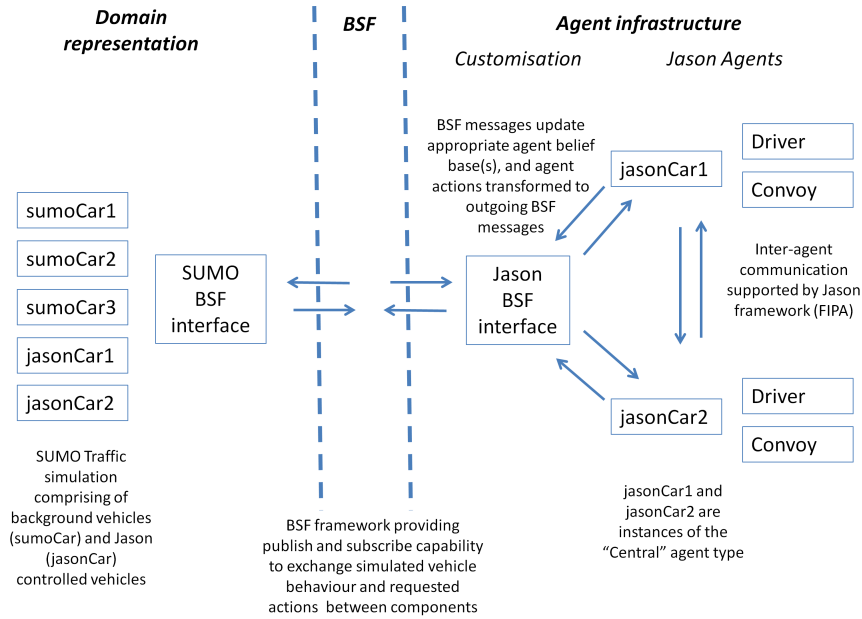


Figure 4-2: Integration of agent infrastructure to domain environment

to individual agents (e.g. if the update is a vehicle in the environment which a specific agent is controlling) or to all agents (e.g. to update all agents of the current environment time), which is managed by the use of the `takenBy` information in the RDF message. In Figure 4-2 this process is shown as the ‘Customisation’ section of the diagram, with an example of two Jason agents, `jasonCar1` and `jasonCar2`, shown. These two agents have each started further agents, of driver and convoy types, and here standard Jason agent communication takes place rather than these agents interacting directly with other BSF components. In the left-hand section of Figure 4-2 the domain representation is shown, in this example showing a number of background (`sumoCarN`) vehicles along with two Jason controlled vehicles (`jasonCar1` and `jasonCar2`). These vehicles populate the corresponding Jason agents belief set, making use of the `takenBy` value of exchanged BSF readings to indicate the origin of the reading, which the Jason BSF interface then uses to update the appropriate agent.

Following the development of the environment representation as sensor input to the agents, similar effort is required to provide agents with the action output capability of Figure 4-1, in order that any desired action from the agent (e.g. stop the vehicle) can be transmitted to the corresponding vehicle over the BSF framework. For this, we extend the Jason `AgArch` (Agent Architecture) class to become `VehicleArch`, a publisher to the BSF architecture which creates appropriate RDF messages based on an agents request. Whilst we consider the operation of a vehicle to be broken down such that it is performed by multiple agents (discussed in more detail in the following Section 4.1.2), we elect to only have one central agent (in Figure 4-2 represented as `jasonCar1` and 2) extended by this specific class, acting as the interface between

a vehicle’s set of agents (e.g. Driver and Convoy agents) and the larger environment. We consider the problems of observability of data and appropriate communication levels from the problem statements of Section 3.2, and assess the trade-off regarding how much data should be published. We create outbound requests from a vehicle, packaged in an RDF indicating it originates from that specific vehicle, and allow the rest of the agents to communicate via an appropriate inter-agent format such as KQML or FIPA (as discussed earlier in Section 2.2.1 and indicated in Figure 4-2).

The issue remains of how much internal system data should be published to the framework, and we attempt to strike a balance with some additional information being published (regarding useful mind-state performance information e.g. number of beliefs), but not every single possible piece of data in order to avoid saturating the network and subscribers. The more general topic of how much to publish, and how to know ahead of time what may be required, is revisited in the further work discussions in Section 7.3.1.

### Agent Definition

Following on from the effort to integrate the agent platform with the BSF infrastructure, such that agents can be situated in the BSF environment, we need to define what agent types and functionality will be created. We aim to devolve the overall management of the vehicle into a number of intelligent agents, and utilise the strengths of agents in being reactive, pro-active and social, where some higher level task (e.g. follow vehicle X) can be composed of sub-goals for a variety of agents to perform. This may also assist in the problem statement of Section 3.2 concerned with observability of the reasoning process, by improving the traceability of an observed event (e.g. vehicle stops) through a chain of agent interactions (e.g. driver agent reasoning that it was about to collide with another object).

In Figure 4-2 we introduced the notion of domain representation separated from the agent infrastructure, with a number of agents residing on the agent infrastructure, responsible for managing the vehicle entity in the targeted domain. Here, we seek to delegate aspects of the vehicle management to a number of agents, which can ask the central agent to request an action from the vehicle if required. This also provides a mechanism to extend to further scenarios if required, by adding an additional agent to provide the required new functionality (e.g. a fuel management agent). We explore the use of three agents which have been constructed to explore the scenarios of Chapter 5, responsible for the following activities:

1. *Central Agent:* This acts as a coordinator between the external environment and the team of agents local to each vehicle, for example passing on agent requests to the remote vehicle and feeding back received updates. This is to allow the agent to de-conflict multiple updates should they occur (e.g. one agent requesting the vehicle speeds up while another requests it slows down).
2. *Driver Agent:* This agent is responsible for the navigation and movement of the vehicle. Such activity has been abstracted away from a low level of control, instead using a more

generic set of commands such as `moveTo(X, Y)`, or loading routes and moving through a series of waypoints.

3. *Convoy Agent*: This agent handles convoy management functionality such as exchanging information with other vehicles ahead of behind it in a formed convoy, and requesting actions from the driver agent as required to perform convoy activities.

The specific uses of these agents is presented in Chapter 5 where they are discussed in conjunction with developed scenarios.

### Agent capabilities

In order to provide a reference source, and awareness of agent construction for the scenarios that follow in Chapter 5, we now present a selection of core agent functionality (i.e. plans, reactions to events) for the `central`, `driver`, and `convoy` agents. This also provides an insight into how knowledge exchange and communication occurs within the agent platform between agents, and how delegation of certain activities is managed.

Firstly, we detail `central` agent capabilities, where this agent is conceptually the gateway between a remote entity and other agents managing aspects of that entity. A key piece of information this agent receives from the environment is a `info` belief, containing a number of vehicle parameters with essential data being `PosX`, `PosY`, `PosZ` for the vehicle location and `Speed` for the vehicles speed. This particular belief illustrates an issue regarding the temporal nature of beliefs, as a vehicle can only be at one position, and on receipt of new position information, all prior beliefs are no longer true and should be expired. However, these beliefs may serve a useful purpose (e.g. to reason about where the vehicle *has* been) but by discarding them, we lose that opportunity. In general terms, this brings us back to problem statements of Section 3.2 regarding understanding and how much to communicate: should the agent attempt to hold on to everything it has ever known, for some potential future reasoning? This topic is revisited both in Section 6.1 where the impact of retaining a high number of beliefs is discussed, and Section 7.3.3 as a further work consideration.

The handling of `info` beliefs received via the BSF framework is initially performed within the `XMPPWorld` Jason environment class, which removes any previous position belief within the `central` agent, and so causes a new belief `info` to be registered within the `central` agent. The agent's handling of this is shown in Listing 4.2, which shows that if the `driver` and `convoy` agent have started (during which they inform the `central` agent of their names) then the `central` agent tells these agents of the newly received `info` belief. This allows those agents to trigger any required plans based on a new `info` belief, rather than having constantly to query the `central` agent, i.e. they are event driven.

Most of the remaining `central` agent functionality provides a mechanism for the `driver` and `convoy` agents to request actions (i.e. to travel at a given speed, to stop, and to move). Whilst this provides the opportunity for the `central` agent to act as a mediator over final action selection (i.e. if there is a conflict between requests from other agents), this capability was not

#### Listing 4.2: Central agent info belief

```
+info(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo) :
  convoyAgentName(CName)
  & driverAgentName(DName)
  //check agents have registered back their names i.e. have started
  & driverName(NameD)
  & convoyMemberName(NameC)
  <- .send(CName,tell,convoyMemberTempInfo(PosX, PosY, PosZ, Health, Speed,
    ChassiHeading, AuxInfo));
  .send(DName, tell,updatedTempPositionHolder(PosX, PosY, PosZ, Health, Speed,
    ChassiHeading, AuxInfo)).
```

---

#### Listing 4.3: Driver agent tempInfo action

```
@upBel[atomic]
+updatedTempPositionHolder(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo
) : true
  <- .abolish(entityUpdate(_,_,_,_,_,_));
  +entityUpdate(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo);
  .abolish(currentRequestedSpeed(_));
  +currentRequestedSpeed(Speed);
  .abolish(updatedTempPositionHolder(_,_,_,_,_,_)).
```

---

#### Listing 4.4: Driver agent entityUpdate action

```
+entityUpdate(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo) :
  currentRequestedSpeed(SpeedVal)
  & started(Owner)
  & collisionMinTime(CTime)
  & geom.convert(SpeedVal,CTime,ChassiHeading, PosX,PosY,PosZ,XFIN,YFIN,ZFIN)
  <- .send(Owner,achieve, updateColPred(PosX,PosY,PosZ,XFIN,YFIN,ZFIN,
    ChassiHeading)).
```

---

used in the scenarios in this work, and so we move on to review the capabilities of the driver and convoy agents.

The driver agent is intended to provide a generic navigation capability to be used in the scenarios presented later in Chapter 5, where the central agent delegates the management of this task, but in keeping with the desire to maintain a generalised solution, this agent could be replaced with an alternative, or delegate further to additional agents. From the earlier Listing 4.2, the central agent tells the driver agent an info belief, and the initial driver agent handling of this is shown in Listing 4.3, which comprises of abolishing any previously held beliefs of position and speed, and re-creating these based on the newly received data.

#### Listing 4.5: Driver speed plan

```
+!setSpeed(NewValue) : currentRequestedSpeed(VOld)
  & started(Owner)
  & standardSpeed(V)
  <- .print("setting new target speed to ", NewValue);
  +currentRequestedSpeed(NewValue);
  .abolish(currentRequestedSpeed(VOld));
  +standardSpeed(NewValue);
  .abolish(standardSpeed(V));
  .send(Owner, achieve, chosenSpeed(NewValue)).
```

---

#### Listing 4.6: Driver distance-speed beliefs

```
distanceRemaining(0, 27.99999999, arrivedAtDestination).
distanceRemaining(28, 10000, cruise).
```

---

The driver agent holds its belief of current position as `entityUpdate`, which as well as being used in a number of other locations within the agent, is also used to update the vehicle's predicted collision volume. This collision volume determines if the vehicle is at risk of a collision (and triggers an appropriate reaction if so), but in more general terms it highlights the problem statement of Section 3.2 regarding the observability of an agents decision making process. The spatial details of this volume are published to the BSF framework, which allows the 3D component to display the volume, giving a human performing debugging the chance to see real environment data (e.g. the vehicle shown at its current location) augmented with internal agent mind state data (i.e. the volume which the agent believes is its current collision zone). This process is shown in Listing 4.4, where an external Java function `geom.convert` is used to calculate the geometry of the predicted collision volume based on the vehicle position, speed, and a `collisionMinTime` belief which represents a reaction time, in order to calculate the volume as the space likely to be occupied over the next `collisionMinTime` seconds.

The driver agent handles two main requests from the central agent: to achieve a given speed, and to move to a given destination. The deliberation is performed by the driver agent, but the action into the environment is called by the `centralAgent` (as shown in Figure 4-2, the `centralAgent` acts as a bridging agent into the environment). The speed plan is shown in Listing 4.5, updating the driver's belief as to what speed it should be travelling at, and asking the central agent to bring about this speed (with the central agent again providing the bridge mechanism due to it having the `AgArch` extension detailed in Section 4.1.2). The speed value is key to the move to location functionality of this agent, based on the beliefs shown in Listing 4.6, where if the the distance to the desired location is less than 28m, this would result in `arrivedAtDestination`, if the distance is above this then `cruise`.

Whilst the cruise plan simply requests the central agent to instruct the vehicle to

Listing 4.7: Driver agent arrived plan

```
+!arrivedAtDestination : started(Owner) <-  
  .send(Owner, achieve, chosenSpeed(0));  
  .abolish(currentRequestedSpeed(_));  
  +currentRequestedSpeed(0);  
  +arrived(yes);  
  +isStopped(true);  
  .send(Owner, unachieve, requestTurnToAngle(_));  
  .drop_desire(moveToKnownPosition);  
  .abolish(desiredXZ(_, _));  
  .drop_event(moveToKnownPosition).
```

---

Listing 4.8: Driver move plan

```
+!moveToKnownPosition : entityUpdate(X,_,Z,_,_,Heading,_)  
  & usingVehicleSim(sumo)  
  & desiredXZ(X1,Z1)  
  & started(Owner)  
  & DX=(X-X1) & DZ=(Z-Z1)  
    & test.test1(X,X1,Z,Z1,Heading,R2)  
  & D1 = math.abs(DX)+math.abs(DZ)  
  & distanceRemaining(MIN,MAX,GOAL) & MIN <= D1 & MAX >= D1  
  <-  
  .wait(250);  
  .send(Owner, achieve, requestMoveTo(X1,Z1));  
  !GOAL;  
  !!moveToKnownPosition.
```

---

travel at the value of `currentRequestedSpeed`, the `arrivedAtDestination` plan shown in Listing 4.7 is more complex. In this case, the plan has to tidy up a number of beliefs and plans which may have been active, but have now been met or are no longer required, and to ask the central agent to request that the vehicle comes to a stop (i.e. set speed to zero).

The final driver agent plan we discuss is shown in Listing 4.8, which provides a generic capability to move the vehicle to a given `desiredXZ(X1,Z1)` position. Two versions of this plan were created

(i) to support integration with the simple XMPP vehicle model (where the driver agent has to set the angle of the vehicle by asking the central agent to `requestTurnToAngle(Angle)`) and (ii) to use with the SUMO traffic simulator (which is the version shown in Listing 4.8). As the traffic simulator handles vehicle navigation based on route following rather than lower level vehicle control such as turning the wheel to achieve a given angle, a `usingVehicleSim(sumo)` belief is used to ensure the correct plan is called. Whilst this is perhaps a less eloquent approach from the view of plan re-use, it provides a useful finding in the desire to maintain a generalised solution, as with some small modifica-

Listing 4.9: Convoy info plans

```
@upCMBel[atomic]
+convoyMemberTempInfo(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo) :
    true <-
        .abolish(convoyMemberInfo(_,_,_,_,_,_));
    +convoyMemberInfo(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo);
    .abolish(convoyMemberTempInfo(_,_,_,_,_,_)).

+convoyMemberInfo(PosX, PosY, PosZ, Health, Speed, ChassiHeading, AuxInfo) :
    started(Owner)
    & vehicleBehind(VehBehind) <-
        .send(VehBehind, tell, vehAheadTempInfo(PosX, PosY, PosZ, Health, Speed,
            ChassiHeading, AuxInfo)).
```

tions the driver agent is now able to interact with two different simulation implementations. In both plans, the `distanceRemaining` belief is called with the calculated values of distance remaining, allowing the plan then to request the corresponding speed.

Having introduced the fundamental driver agent functionality, we now move on to present the core convoy agent details. This agent has the responsibility for managing convoy behaviour, for scenarios where vehicles are travelling in a formation to some destination (specific details follow later in Chapter 5). We propose two key requirements when vehicles are in convoy formation, firstly for all to follow the same route, and secondly to maintain a steady gap to the vehicle ahead, and it is these functions that the `convoy` agent focusses on.

Firstly, we saw earlier in Listing 4.2 that the central agent informs the convoy agent of `convoyMemberTempInfo`, and in Listing 4.9 we see a similar approach to the driver agent; any old beliefs regarding the position are removed, before a new belief `convoyMemberInfo` is added. It is also shown in Listing 4.2 that this only triggers an action if the convoy agent holds a belief `vehicleBehind(VehBehind)`. This is used in the perception based convoy scenario outlined in Section 5.2, where convoy agents push frequent low level information as a means of functioning as a convoy.

The alternative to pushing new position data to a vehicle behind, is for a vehicle to request it (i.e. a pull approach), which is the approach explored in an alternative strategy detailed in Section 5.2 to reduce the communication burden. This relies on the convoy management plan shown in Listing 4.10, where the `convoy` agent asks the `convoy` agent of the vehicle ahead, for that vehicles current location. The returned location is then used to tell driver agent of the desired location, with the `convoy` agent then deciding how to manage the distance to the car ahead via the `manageDistance` plan.

The `convoy` agent `manageDistance` plan is shown in Listing 4.12, where the distance to the position of the vehicle ahead is calculated, and then the belief set shown in Listing 4.11 is used to find an appropriate action, comprising of slowing down (if the gap is too small),

Listing 4.10: Convoy management plan

```
+!convoyMgmtPlan : vehicleAhead(VehicleAhead)
  & myDriverName(DriverName)
  & amMoving(false) <-
    .send(VehicleAhead, askOne, convoyMemberInfo(X,Y,Z,_,_,_),convoyMemberInfo(X1,
      Y1,Z1,A1,B1,C1,D1));
    .send(DriverName, achieve, clearArrivedState);
    .send(DriverName, tell, updatedTempMoveXZ(X1,Z1));
    .send(DriverName, achieve, moveToKnownPosition);
    !manageDistance(X1,Y1,Z1);
    .wait(1500);
    !convoyMgmtPlan.
```

---

Listing 4.11: Convoy distance behaviours

```
distanceToCarInfront(0, 10.9999999999, applyBrakes).
distanceToCarInfront(11, 34.9999999999, slowDown).
distanceToCarInfront(35, 45.9999999999, standardSpeed).
distanceToCarInfront(46, 200.9999999999, speedUp).
distanceToCarInfront(201, 1000, lostMyLeader).
```

---

Listing 4.12: Convoy distance ahead plan

```
+!manageDistance(DPosX,DPosY, DPosZ) : convoyMemberInfo(MyX, MyY, MyZ, Health,
  Speed, ChassiHeading, AuxInfo)
  & myDriverName(DriverName)
  & DX=(DPosX-MyX) & DY=(DPosY-MyY) & DZ=(DPosZ-MyZ)
  & DTotal = math.abs(DX)+math.abs(DY)+math.abs(DZ)
  & distanceToCarInfront(MIN,MAX,GOAL) & MIN <= DTotal & MAX >= DTotal
  <- .send(DriverName, achieve, GOAL).
```

---

speeding up (if the gap is too large), doing nothing (if the gap is in an acceptable range), or if the gap is too large, calling a `lostMyLeader` plan which in the current implementation stops the vehicle (as something unexpected has occurred).

The final convoy capability presented is used in the scenario of Section 5.3, where to investigate problem statements regarding building awareness and understanding, as well as alternative approaches to communication, convoy movement is based on the use of a route (consisting of a set of waypoints) rather than the lower level percept exchanges of the vehicle ahead position. Two plans to handle this are shown in Listing 4.13, where when a convoy agent uses the `passBackWaypoints` plan to pass back its set of way points to the convoy agent of the vehicle behind it, it then finishes by telling that same agent to `followWaypoints`. For a convoy agent to achieve this, it passes the locations to the vehicle driver agent which treats



Listing 4.13: Convoy waypoint plans

```
+!passBackWaypoints : started(Owner) & vehicleBehind(VehBehind) <-  
  .print("passing waypoints back along convoy members to ", VehBehind);  
  .findall(wayPoint(A,B,C), wayPoint(A,B,C), Val);  
  .send(VehBehind, tell, Val);  
  .wait(1000);  
  .send(VehBehind, achieve, followWaypoints).  
  
+!followWaypoints : myDriverName(Driver) & .count(wayPoint(_,_,_),N Ways) <-  
  .print("pushing ",N Ways," waypoints to ", Driver);  
  .findall(wayPoint(A,B,C), wayPoint(A,B,C), Val);  
  .send(Driver, tell, Val);  
  .wait(1000);  
  .send(Driver, achieve, followWaypoints).
```

---

these as a series of requestMoveTo positions and draws on the moveToKnownPosition capability shown earlier in Listing 4.8.

This concludes the details of the intelligent layer implementation, which is based on supporting investigation into the problem statements set out in Section 3.2 whilst implementing sufficient capability in the domain of intelligent vehicles to support the scenarios which follow in Chapter 5. Key features are the observability of data and a generalised distributed approach, achieved through the adoption of the BSF framework as the communication medium between the agent framework and the remote environment.

### 4.1.3 Entity representation

Whilst a number problem statements of Section 3.2 consider general issues facing socio-cognitive systems, there is a specific statement to demonstrate any proposed solution against a problem domain, and for this we focus on the domain of intelligent vehicles. As discussed in the introduction of Section 4.1, we adopt an evolutionary approach to the development, and first considered the problem statements of developing understanding and creating observability of the reasoning process and information flow. As such, we first explored an existing solution where an intelligent agent platform had been integrated into a vehicle domain, to assess how we can transition the findings of Section 4.1.2 to the intended target domain. From the understanding developed in this effort, we then focussed on two efforts: a generalised integration with the intelligent layer, and the provision of the environment (as per Figure 4-1) via a distributed component. This seeks to address the problem statements of retaining a generalised solution, in combination with a distributed systems approach. As this environment representation should therefore be suitably de-coupled, we present the findings of introducing real devices to the framework in order to test the generalised approach and establish what different communication profiles might look like, to inform our view regarding the problem statement regarding quantity and appropriateness of communication. Finally, we move back into the specific domain of intelligent

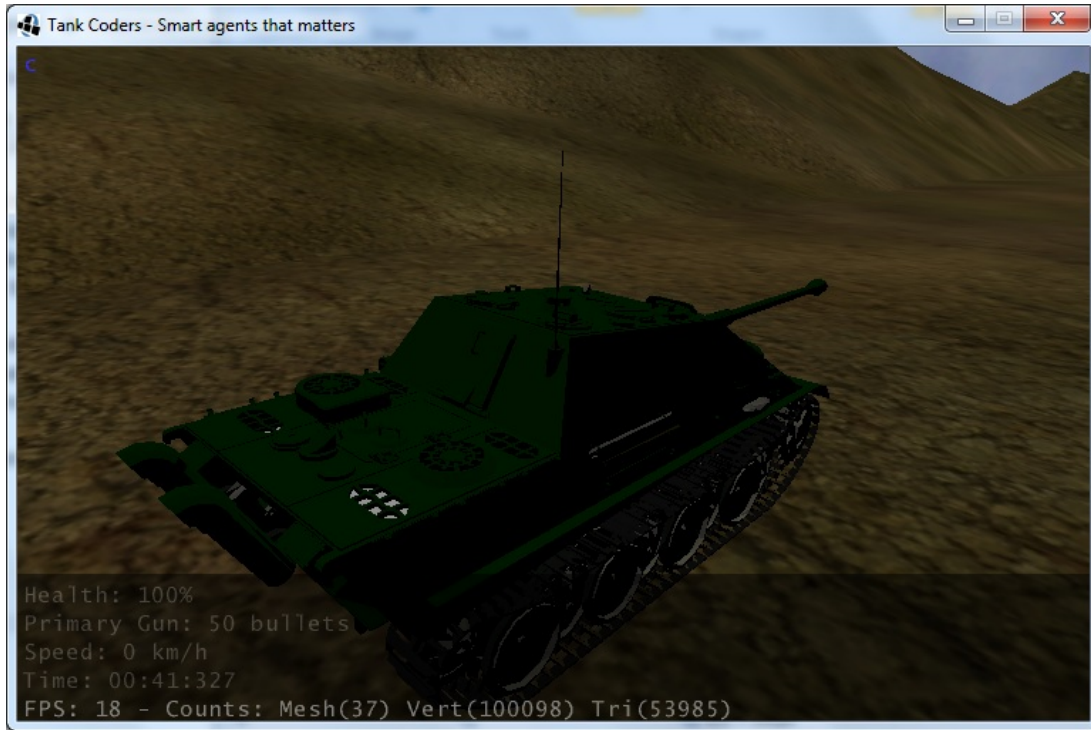


Figure 4-3: Tankcoders example view

vehicles, and present details of the integration of a vehicle and traffic simulator which provides the final environment in which the intelligent agents operate.

### TankCoders

The first experimentation took place with a platform called TankCoders (presented in [Fronza \[2008\]](#)). This is an adversarial agent based tank simulation where agents have been implemented in the Jason agent platform, and jMonkeyEngine ([Kusterer \[2012\]](#)) is used both as the 3D visualisation tool and to model vehicle physics. The jMonkeyEngine interface can be used by a human to control a tank against an agent based opponent, and supports mixed human-agent players using distributed connections over a network. An example of the view from this application is shown in Figure 4-3, which can be compared to later images to contrast visual components and scenario contexts.

The key information communicated to Jason agents is of the form:

```
info(PosX, PosY, PosZ, Health, MainGunBullets, ChassiHeading,
MainGunHeading)
```

with reasonably self explanatory data - position, health, number of bullets remaining, and heading of chassis and gun. Additional key messages include:

```
+onScanTank(TankName, IsEnemy, PosX, PosY, PosZ)
```

when other tank vehicles are detected, and various updates regarding being hit, whether a fired bullet has hit another tank, and so on. Available actions to the agents include `ahead(Qty)` to move forward by `Qty` metres, with `back(Qty)` providing the same for moving backwards, `turnRight(Angle, Direction)` and `turnLeft(Angle, Direction)` to rotate the tank `Angle` degrees and similar actions for firing the gun, turning the gun, and so on.

This set of available updates and actions provided a reasonable basis on which to experiment with vehicle control via Jason agents. Along with this, a number of useful findings arose from investigation into the TankCoders solution, as the approach and component selection (e.g. distributed, intelligent agents, controlling some entities) aligns with the requirements put forward earlier in Section 3.4.

Subsequently, we tried to build on TankCoders, moving from adversarial to cooperative agents and exploring information exchange in a convoy of TankCoders vehicles, and implementing the agent driving team outlined earlier in Section 4.1.2. TankCoders provided a useful prototyping facility, as key agent behaviours could be developed against an existing package, and since they are built in Jason, they remain transferable both to other simulation components (i.e. if TankCoders is replaced with something else) and to other domains (as a `moveTo` plan could be applicable to other platforms, for example a cleaning robot).

The TankCoders experiments lead to the finding that closely coupled components running on the same hardware could encounter performance issues, with both the 3D engine and Jason struggling under load (greater focus on this follows in Section 4.2). In general terms, this seems related to the coupling of a computationally expensive 3D engine, physics representation, and agent platform, all on the same physical computer, with little control over message rates, fidelity, etc. This provides additional motivation to pursue the problem statement of Section 3.2 concerned with adopting a distributed approach, in order to spread the load across multiple computers.

## XMPP Vehicle

Whilst TankCoders provided a good platform for early investigation and prototyping of agents, to address the problem statements of Section 3.2 there is a need to de-couple components and develop a distributed systems approach. This also aids in exposing communication between components in order to support the observability requirement, in forcing consideration for packaging information in some suitable format for distribution, i.e. the *what* aspect of communication. It also brings the benefit that computational requirements of software can be spread across hardware, for example the 3D software running on a PC with suitable graphics capability, compared to the agent platform which requires CPU but no significant graphical power.

This effort was planned in two phases, built around using the Bath Sensor Framework (BSF) model rather than the integrated approach of TankCoders. The BSF was discussed earlier in Section 4.1.1, and was introduced at this point to separate the Jason agents from the vehicle representation and visualisation. Phase one would replace the vehicle representation

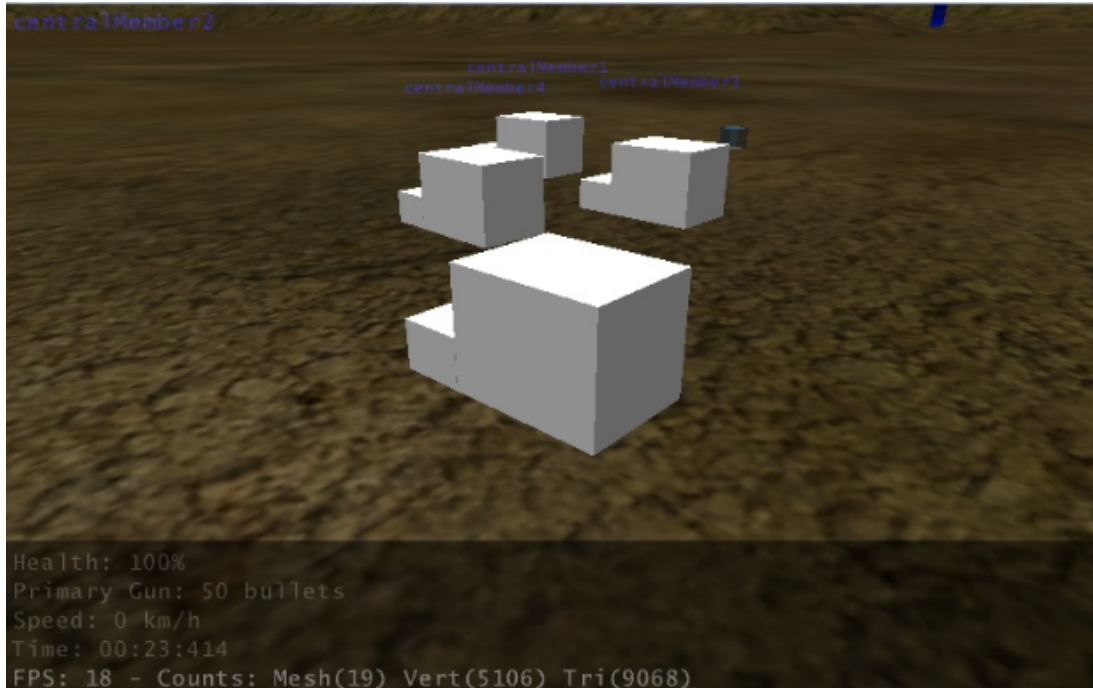


Figure 4-4: XMPP vehicles from phase one migration

with instances of an `xmppVehicle`, a very simple model but running externally to the work contained in `TankCoders`. In phase two, the agent platform would be split out, communicating via BSF to the XMPP vehicles, with a separate component of a 3D visualiser available in order to view the simulation. At this point, whilst the apparent capability would be similar to that of `TankCoders`, it has moved away from that solution, becoming now based around the BSF.

The visual result of this first phase of migration can be seen in Figure 4-4, and whilst the only visual clue that something has changed in comparison to Figure 4-3 is the much simpler visual model of a vehicle, the BSF is now providing geospatial information for these vehicles to the `TankCoders` software, with a new `xmppVehicle` component developed in the BSF software repository. In BSF terminology, this is a reasonably simple sensor and publisher, responding to `"http://127.0.0.1/request/setOrientation"` and `"http://127.0.0.1/request/setSpeed"` received messages, and publishing `"http://127.0.0.1/sensors/types#spatial"` messages. The `xmppVehicle` performs an update each second, where given its current heading, speed, and location, it calculated where it would have moved to in that time period, and sets this as its new position, which is then published to the BSF. The driver Jason agent is then able to control the vehicle's movement through the use of `setSpeed` and `setOrientation` requests, and on receipt of new spatial data from the vehicle, calculates an appropriate heading and speed for a given desired destination.

This brought about two useful achievements, that agents were now no longer closely tied to the vehicle they were controlling (i.e. we had refactored this part of the implementation), and

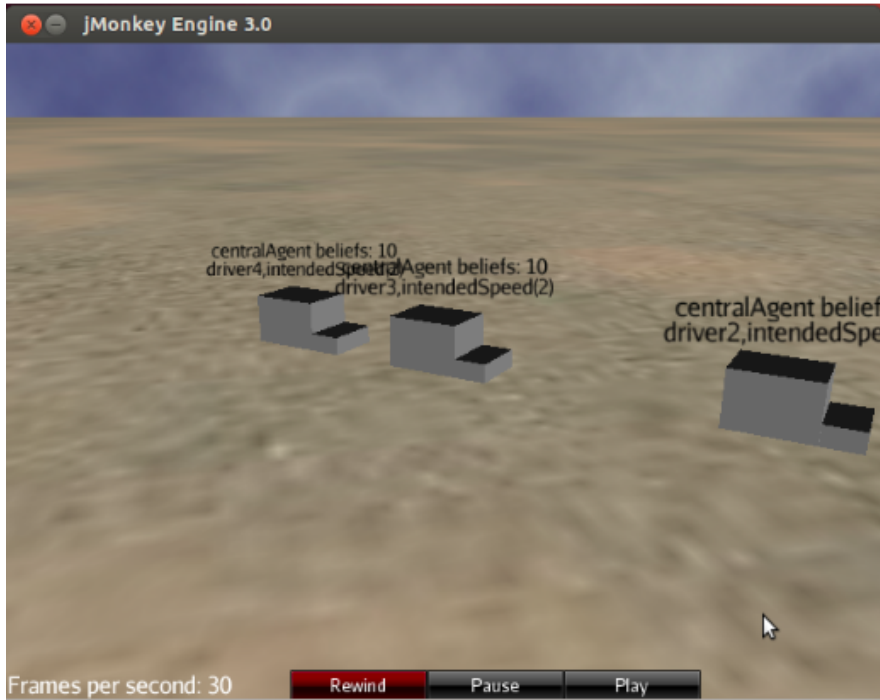


Figure 4-5: XMPP vehicles from phase two migration

that performance (in terms of load on the system) was slightly improved since the physics modelling of the vehicles was no longer being performed in the same application (as the combined Jason and 3D component).

Following this, phase two was implemented, separating the 3D view and the Jason platform. The visual result of this can be seen in Figure 4-5, where although the same 3D vehicle model has been used, the GUI contains a different structure. The implementation within Jason follows that discussed in Section 4.1.2 with customised extensions of Jasons environment class (to process the spatial info received from XMPP vehicles) and agent class (to translate agent actions to setOrientation and setSpeed messages). Additional changes can also be seen in Figure 4-5 where we have begun to address the problem statement of Section 3.2 regarding observability, and here we seek to tackle this by showing both communicated messages, and mind state data of the agent itself. In Figure 4-5 text appears above the vehicles, showing the agents current number of beliefs, and last received message (in this case, 10 beliefs, and an intendedSpeed(2)). This is discussed in more detail in Section 4.1.7, but is mentioned here to show the benefits of this new structure in improving the visibility of agent processes.

### Real platforms

From the problem statements of Section 3.2, we put forward issues regarding quantity of data communication, and demonstrating a generalised distributed system. Having decoupled the





Figure 4-6: Android XMPP Vehicle

agent infrastructure from a target device, we now take the opportunity to explore the feasibility of linking the BSF to real devices, in order to assess i) what challenges this involves i.e. how de-coupled the system really is, ii) factors introduced by real devices, in order to assist in maintaining a generalised solution, and iii) what impact such devices may have on communication, i.e. introducing message delays, sensor noise, and high data rates .

As the assumption was that the vehicle component was now de-coupled from the agent framework, the first experiment seeks to test this assertion by replacing the XMPP vehicle with a real vehicle, and integrate it into the BSF. To reduce the complexity of the effort, a simple remote control car was used, in conjunction with an Android mobile device, as shown in Figure 4-6.

This implementation used the Android mobile device to act as the interface to the BSF, using the same publish and subscribe methods provided by the `xmppVehicle` but now reporting live sensor data. The device was coupled with an IOIO<sup>3</sup> board in order to provide an input/output control over voltage signals, which was then wired to a simple remote control vehicle. As low end models of remote control vehicles simply receive voltage high/low for signals (e.g. move forward, move backward), removing the remote control aspect of these signals and replacing with signals from the IOIO board allows the Android device to take over operation of the vehicle. With this in place, a mapping of BSF published vehicle commands (e.g. turn left) to IOIO control signals (e.g. set pin 2 high) provides the real world control. Whilst this resolves the problem of how to replace the handling of received commands, the issue of how to publish

---

<sup>3</sup><https://github.com/ytai/ioio/wiki>

vehicle position data back to the BSF remains. Typically, Android devices are fitted with number of sensors such as GPS and inertial sensors, and whilst GPS could offer a solution to the location problem, there is the issue that any increase in the potential for unreliable position data requires the capacity for the intelligent agents to handle this. In this case, the use of GPS introduces accuracy problems (as well as requiring line of sight to sufficient satellites). Alternative position calculation based on inertial sensors introduces accuracy issues (discussed in [Woodman \[2007\]](#)) and so that approach is unlikely to resolve the issue. Indoor WiFi tracking could be another approach, with approaches such as Redpin shown by [Bolliger \[2008\]](#) as being able to achieve room level accuracy for a WiFi device.

However, this thread of research was not developed further, in order to avoid being pulled into problems which are not core to this thesis, but useful findings to inform the problem discussions of Chapter 3 were discovered. Whilst the plausibility of connecting the framework to another device was shown, it serves as a reminder of how adopting an idealised domain representation can mask complexity of the problem domain, and provides motivation to find a more suitable vehicle and traffic simulation. However, it does add understanding to how the framework could be integrated into the live domain, and that the distributed approach allows subscribers to BSF messages to implement these as appropriate for their device, whether this be a simulated or real vehicle.

The second experiment was to integrate a quadcopter (Parrot AR.Drone2.0<sup>4</sup>) into the BSF to explore a variation of domain, shifting to air instead of land vehicles, which compared to the previous vehicle investigation now includes a working solution to indoor and outdoor spatial information. In this case, through the use of onboard sensors and camera data ([Engel et al. \[2012\]](#)) position information is available without a dependency on GPS (though GPS is also available for outdoor navigation).

A Java SDK “YADrone”<sup>5</sup> was used to create a BSF bridge component to the quadcopter, which would run on a PC with two network interfaces. One of these interfaces connects to an unmodified AR Drone, and the other interface connects to the local network containing BSF components. The bridge component then translates received BSF messages (indicating an action is required e.g. to take off) to appropriate AR Drone commands (called via the YADrone library), and publishes received telemetry to the BSF middleware.

Whilst the processing of received commands proved straightforward, publishing data to the BSF raised some useful questions. It was found that the data rate sent from the AR Drone was higher than previous experiments, at approximately 300 messages per second. The effect of the ARDrone publishing at this rate is shown in Figure 4-7, where it can be seen that the number of received messages per second is always slightly below the number being published, suggesting that the framework is struggling under this load. Given the question of appropriate levels of communication set out in the problem statements of Section 3, this shifts the consideration that in alternative domains we may have to deal with much higher communication levels, though it

<sup>4</sup><http://ardrone2.parrot.com/>, accessed 30th August 2014

<sup>5</sup><http://vsis-www.informatik.uni-hamburg.de/oldServer/teaching/projects/yadrone/>, accessed 30th August 2014

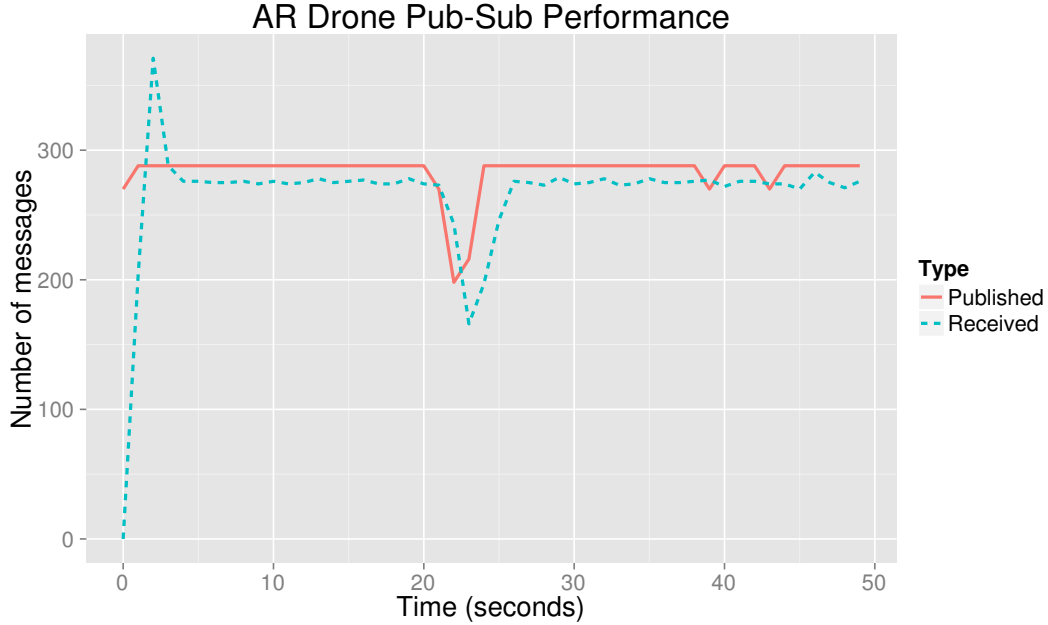


Figure 4-7: AR Drone Pub-Sub rate

also poses the question of whether this much data is required. This becomes a common theme in this work, where different components may require different data rates, but also different information levels (in the SA sense), and whilst adopted the exchange of higher level information may reduce the burden of high message quantity, this may not always be an option (especially in an air domain where entities such as quadcopters may be moving at high speeds, navigating in urban situations).

Whilst performance improvements were developed to handle this (discussed in more detail later in Section 4.2), the question arises as to how much is an appropriate amount of information to send. Another aspect was the handling of video, as the AR Drone transmits its video feed back to the bridge component. In the experimental work here, the video was then encoded and sent to the messaging server where it could be available to any client via an appropriate URL. However, it is feasible that a subscriber may wish to receive this feed directly (e.g. to perform image analysis), but if large numbers of subscribers all request this there will be a significant demand placed on the infrastructure. This raises the consideration, that whilst in this work we consider various communication needs, the assumption is based on message exchange, rather than the challenge of larger data types such as video. However, we leave this aside as an infrastructure issue where ever increasing wireless communication capability may be able to carry such data, and appropriate tuning of the framework (e.g. load balancing) may mean there is no issue in transmitting alternative message payloads such as audio and video.

In summary, these two investigations proved useful in testing how the framework could work with real world platforms, highlighting the issue that even one real world device has the



capability to saturate the message network if a direct connection is created. Either the device has to perform some filtering before publishing data (potentially as simple as a delay between message publication) or on the subscriber side some additional handling has to be performed (such as the approaches discussed earlier in Section 2.1.2) to deduce an appropriate SA level to the subscribing component. Such questions are core to the generalised problems set out in Chapter 3, and echo discussions around situational awareness, i.e. what information needs to be communicated, at what level, and so on,

These investigations also add benefit in testing the assertion that the framework is reasonably agnostic to what problem domain it operates in, be it simulated vehicles, real vehicles or devices from another domain. This is useful reassurance, as just with vehicles moving towards autonomous control (as discussed in Section 2.3.1), there are similar developments happening in the air domain, which relates to the general problem of introducing novel technology put forward in Section 3.1. For example, Amazon seem to be considering the use of quadcopter drones for transportation of items, in a letter to shareholders [Bezos \[2014\]](#) discussing their “Prime Air team” testing a 6th generation of aerial vehicles with design phases underway for an 8th generation. The problem of coordinating a significant number of such drones, in a city airspace, for both safety and speed of delivery, interacting with existing air users, could be a potential application for the approach presented in this thesis.

### Improved simulation

The approach to entity representation has been considered from a number of angles at this point, from the Tankcoders simulated vehicles, to a simple remote vehicle representation suitable for the distributed systems approach, to initial investigation in using real land and air vehicles. This leads to the conclusion that an appropriate level of domain representation is required in order to investigate the problems set out in Chapter 3, but not so complex to become bogged down in domain specifics and lose the generalised solution approach. The simple XMPP vehicles are highly idealised (so vehicle effects such as fuel consumption were not modelled) and also there was no background traffic to interact with, or domain rules (e.g. which side of the road to drive on, when it is permitted to turn). Given the problem statements of Section 3.2, we require a population of vehicles, behaving according to appropriate rules, such that we can construct scenarios which assess individual versus collective issues, and the resolution of problematic situations. Such requirements push us towards finding an appropriate traffic simulation package, to support repeatable, safe experimental scenarios to be constructed.

Initially consideration was given to modelling the required features in the `xmppVehicle` (e.g. implement some model of fuel consumption in relation to speed), though this could quickly divert research effort into building some customised vehicle and traffic simulator, raising concern over the level of expertise required to model such specifics, and also the risk of wasted effort if such software already exists. In order to avoid these problems, the availability of suitable vehicle and traffic simulators was investigated, and two promising software packages were found, MATSim (presented in [Balmer et al. \[2008\]](#)) and SUMO (presented in [Krajzewicz et al. \[2012\]](#)).

These were discussed earlier in Section 2.3.4, and initial assessment suggested SUMO would be a good candidate to provide the required simulation capability and integration with the BSF.

The main requirement is for bi-directional data flow, i.e. information needs to be extracted from a simulation package, but also information from BSF members needs to be able to flow back to the simulation. Without this, any vehicles created outside of the vehicle-traffic simulation will not appear to that simulator, causing a disconnect between what the intelligence layer may be reasoning about and what the simulation is representing and acting upon. However, it seems rare (and reasonably so) that a simulation package would include the capability to handle remote entities when this was not in their original design specification. So, whilst it may be possible to extract required information from the simulation (e.g. vehicle positions, speeds, etc) and publish these to the BSF, integrating control from the intelligent agents to the simulated vehicles proves more challenging.

Fortunately, it was found that SUMO offers sufficient API access to exert reasonable control over vehicles, and even where shortfalls were found, work arounds have been possible. A Java interface (Traci4j<sup>6</sup>) allows control over simulation time steps, the ability to insert vehicles, and to control some characteristics of vehicles (e.g. route, speed, etc). Based on this, a BSF component was developed which calls on the Traci4j library, initiates the simulation, and then converts received BSF messages to appropriate SUMO calls, publishing extracted SUMO data to the BSF for any other components (e.g. 3D view, agent platform, monitoring tools) to consume.

A significant amount of information is available from SUMO, which provides a rich data source to the BSF-based simulation. The benefit has been the ability to create scenarios in both a city (Bath) and motorway (M25) context, exploring across a spectrum of speeds (e.g. low city speed vs high motorway speed), multiple lanes, congestion, and interactions with other artefacts such as traffic lights and variable speed limits. Information on specifics of scenarios constructed in SUMO follow later in Chapter 5.

Interoperability has been maintained by basing the scenarios on OpenStreetMap (OSM) map data (for which there are a number of freely available tools to parse, view and consume in general) used in both SUMO and the 3D viewer (described later in Section 4.1.7). The motorway scenario has also been built around real world flow data, made available from the UK Highways Agency Traffic Flow Database System (TRADS Agency). Although this is only provided in 15 minute samples, and does not provide a lane breakdown of the flow, it does provide four vehicle categories as components of the total flow, and offers a significant improvement to the reliability of scenarios implemented in SUMO.

The key benefit is that the overall scenarios are now tested against a recognised traffic simulation package, improving the confidence of others in the validity of the work and allowing much more advanced scenarios to be constructed.

---

<sup>6</sup><https://github.com/egueli/TraCI4J>, accessed 30th August 2014

#### 4.1.4 Institution Framework

Of (the many) problems faced in artificial socio-cognitive systems, two issues are put forward in the problem statements of Section 3.2 pertaining to the social aspects: resolving problematic situations, and balancing individual pursuits against the greater social welfare. In these issues, we see a need for some external body, with appropriate empowerment to bring about the resolution of challenging situations, and to influence the behaviour of intelligent agents in order to restrain their total pursuit of own desires when the wider needs of the population may be to the contrary. These issues relate to the earlier discussions in Section 2.2.2 regarding the work of Ostrom [1990] and the issue of the “tragedy of the commons”, and we seek to use an institution framework as a means to govern the individual pursuits in such cases.

The topic of institutions was introduced in Section 2.2.2 as a means of formally specifying a set of rules to govern a given situation, and in our case to provide some management of agent autonomy. Earlier work presented in Lee et al. [2013] has integrated an institutional framework (based on the work of Cliffe et al. [2006]) into the Bath Sensor Framework, and demonstrated its use in assisting Jason agents with social reasoning. We seek to expand upon this work in order to address the problem statements set out in Section 3.2, and outline our use of institutions before providing details of their use in specific scenarios later on in Chapter 5.

Firstly we provide some formal definitions regarding the use of institutions, both to define the process and terminology used, and relate them to the process and architecture used in this research. However, we limit this to focussing on terms used within this work: rather providing an in-depth review the entire area, instead we attempt to distil the extensive definitions and work put forward by Cliffe et al. [2006] in order to provide sufficient terminology to explain the integration with the BSF, and use in the scenarios of Chapter 5.

As a component within the BSF framework, significant emphasis is on *events*, which comprise of two sub-categories, the first being *observable events*, that is, events which are external to the institution and take place in the physical world (e.g. a vehicle collision). The second type are *institutional events* which relate to events generated by an institution, having meaning within that context (e.g. driver committed a crime). The work of Searle [1995] provides a rationale for the process for generating institutional events by *Conventional Generation*, where external observable events are able to *Count As* the occurrence of an event in a different context. This allows us to translate from the first context of the physical world, into the second context of the institution and generate institution events.

As *events* occur, the *institutional state* changes from an initial state  $S_0$  to discrete evolving states  $S_N$ , where each state  $N$  contains a set of *institutional fluents* that characterise the state. A fluent denotes a fact that is true when present and false when absent. Of the four kinds of institutional fluents, three are referred to as *normative fluents*; these comprise of *power*: whether an event can be brought about), *obligation*: that an event must happen before some other event), *permission*: an event that is a valid occurrence. The fourth are *domain fluents*, which represent facts specific to the domain in which the institution is acting.

Finally, there is the formal process of affecting the institutional state based on the occurrence

1. An InstAL specification file is created (e.g. of type in Appendix A) which defines the institution behaviour
2. This is compiled to an Answer Set Program, which is then entered into an ASP solver along with the external event(s) of interest
3. The solver takes both of these as inputs, and computes a results file
4. This results file contains the output of institutional states over N steps
5. Items of interest such as permissions and obligations can then be extracted from this result file

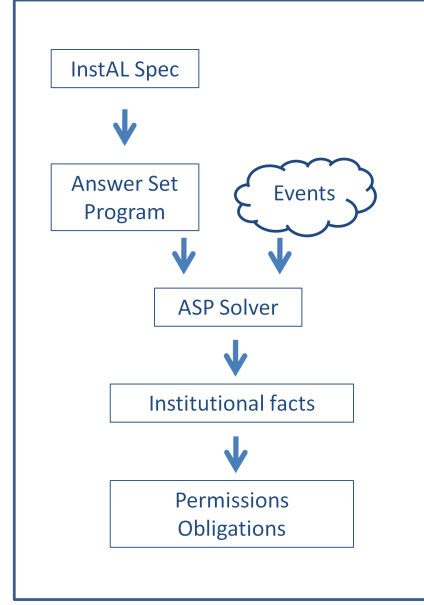


Figure 4-8: Institution information flow

of events, which is achieved through *generation rules* and *consequence rules*. Generation rules determine how the occurrence of event(s) should create any institutional event(s), whereas consequence rules determine the initiation or termination of fluents.

Based on these concepts, the definition of an institution can be put forward as  $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, S_0 \rangle$ , comprising of institutional events ( $\mathcal{E}$ ), fluents ( $\mathcal{F}$ ), consequence rules ( $\mathcal{C}$ ), generation rules ( $\mathcal{G}$ ), and the initial set of fluents ( $S_0$ ).

This approach can be extended to cover interaction between multiple institutions, where one institution is able to cross generate and initiate fluents in another. In Li et al. [2013a,b], this concept is explored as a means to identify legal conflicts which may arise between two institutions (using a case study of EU privacy law versus US Surveillance law), through the development of a *bridge institution*. This bridge institution defines the relationship between institutions, and the power of the generation and consequence rules that one institution has over another, allowing internal institutional states to affect other institutions (which is explored in Scenario 5.7 as a capability to bring multiple institutions together to cooperate in the resolution of a problem).

Following this formal definition, we set out a representation of information flow for three cases i) use in a standalone institution sense, ii) the use of a single institution within the BSF, and iii) multiple institutions operating within the BSF. For the case of multiple institutions, we discuss two variations: multiple instances of single institutions with no direct interaction between institutions, and multiple institutions with interaction occurring between institutions.

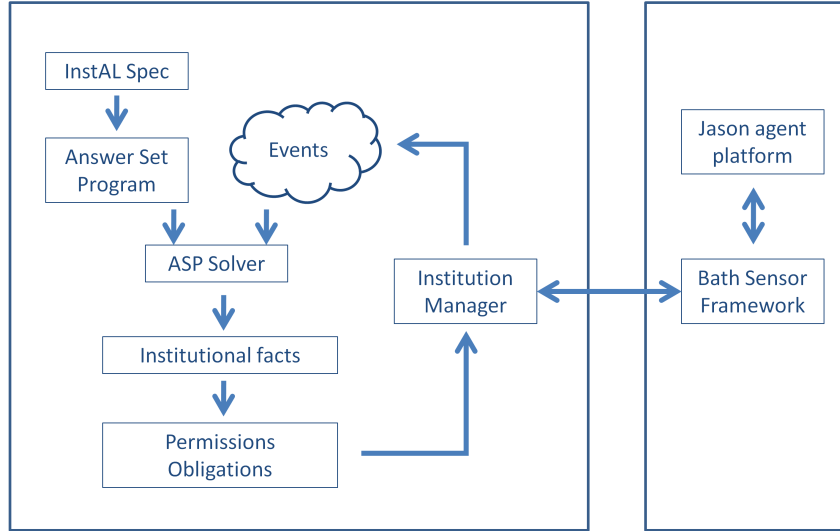


Figure 4-9: Institution integration into the BSF

In Figure 4-8 the standalone use of an institution is presented, along with a brief description of each step in the process. The first step introduces the InstAL specification as a key component in defining institution behaviour. This contains the definition of fluents, consequence and generation rules, and so formally defines institution behaviour. As indicated in Figure 4-8, this information is then turned into a computational representation, and when combined with a current set of events and fluents, generates a set of institutional facts, from which items of interest such as permissions and obligations can be extracted.

Whilst the approach of using institutions in a stand-alone mode as shown in Figure 4-8 is perfectly valid for some uses, we are interested in using this as a real-time service, in conjunction with the BSF, to provide assistance for the genre of problems outlined in Section 3.2. To accomplish this, the structure shown in Figure 4-9 is adopted, where a specific BSF component referred to as the “Institution Manager” provides additional control and integration. The institution manager is responsible for instantiating the institution, and using any relevant BSF messages as *observable events*. The InstAL specification is translated to a logic program based on Answer Set Programming (ASP), in order to allow computational reasoning (of the formal model set out earlier). This allows external events to be reasoned about in combination with the ASP model, in order to determine the new institution state. Depending on what these external events are, and the institution specification, there may be a number of outcomes both internal to the institution and also with external relevance (i.e. the issuance of obligations). The role of the institution manager is to create appropriate BSF messages based on such externally relevant institution events, which can then be received and acted upon by other BSF components.

We draw on the scenario of Section 5.6 to provide an example of how this flow works for the case of a single institution being used. In this scenario, the process flow is started with

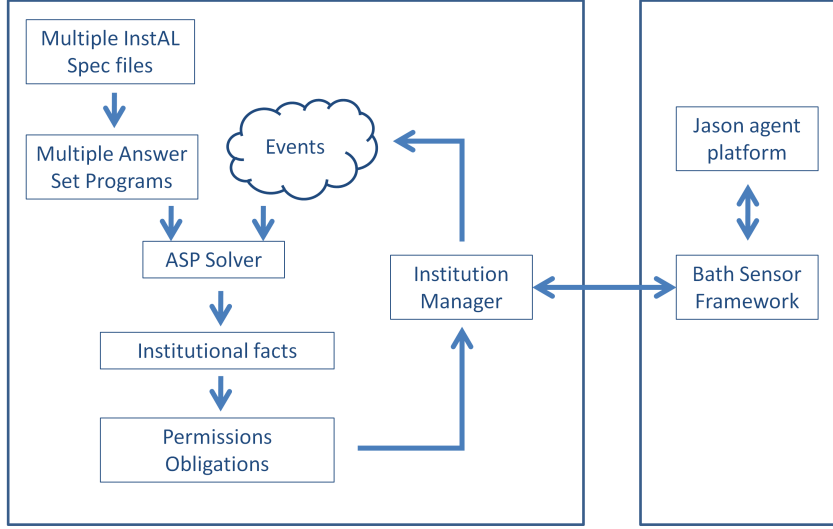


Figure 4-10: Multiple institutions integrated into the BSF

an emergencyBrake event, after which the institution manager instantiates a Variable Speed Limit (VSL) institution to manage the situation. The institution manager translates received BSF messages into appropriate external events and initial fluents, which are combined with the answer set program created from the institutional InstAL definitions. The institution manager then takes any obligation(s) resulting from this process and publishes these to the BSF (in this example, of the form `slowDown(VehicleName, Speed)`, which BSF recipients then process accordingly.

Considering the problem statements of Section 3.2, we see this capability as a means to provide assistance to intelligent agents where they lack the capability to handle (or awareness that they need to handle) events relating to social convention, and where an agent’s focus on pursuit of its own goals may be to the detriment of the wider society. With our approach of observability of message exchange, the institution manager is able to monitor events taking place in these scenarios, calling the appropriate institution(s) where necessary and relaying the outcome of that institution resolution to the intelligent agents (and any other BSF components which have subscribed to that data stream).

This approach can be taken further, in the use of multiple institutions interacting with BSF components. One possibility is to use multiple instances of a single institution manager (shown in Figure 4-9), to allow more than one institution to have influence over BSF members where there is no direct interaction between institutions required (e.g. one may issue speed obligations and another may issue obligations to not park in a certain area). However, there is an additional case where interaction is required between institutions in order to share fluents and to transition through shared states. This approach is shown in Figure 4-10, where in contrast to the earlier Figure 4-9 multiple institution InstAL definition files (including a bridge

definition) are loaded by the institution manager, allowing interaction between institutions in response to the events received via the BSF.

The multiple institution approach from Figure 4-10 is explored in the scenario of Section 5.7, where the process is triggered by a vehicle collision event occurring. On detection of this (through analysing BSF messages) the institution manager instantiates the appropriate institutions, translating the received BSF message into an appropriate external event (along with any initial fluents) to be combined with the answer set program created from the institutional InstAL definitions. The institution manager then takes any external events this has generated, and produces appropriate messages to be published to the BSF (in this example, of the form `mergePairing(VehicleToMerge, VehicleToPermitMerge, CrashedVehicleName)`), which BSF components then process accordingly. In this example, the intelligent framework passes the message to the relevant vehicle agents (i.e. `VehicleToPermitMerge`) where they implement an appropriate `mergePairing` behaviour.

These various approaches to the integration of institutions within the wider BSF framework support the use of single and multiple interacting institutions used in a number of scenarios discussed later in Sections 5.4, 5.5, 5.6 and 5.7.

#### 4.1.5 Area Of Interest

From the problem statements of Section 3.2, the issue of finding the balance between quantity and appropriateness is put forward, where the topic of Situational Awareness is proposed as a means to consider data as being low level (i.e. basic perceptions) through to higher level (i.e. comprehension of events, projection of future states of the environment). However, given the distributed systems approach put forward in those problem statements, there is the consideration that not all framework members may be able to handle the rate at which various information streams are published, which relates to the issue of impedance matching introduced earlier in Section 3.4.1. For example, the agent framework may need higher level information but lack the capability to generate these itself (e.g. lack of compute cycles due to the deliberative agent process), and may at a more fundamental level be incompatible with a task (e.g. to perform dedicated sensor fusion which would otherwise take place on specialised hardware). It is from this that we identify the need to develop a component for the framework which is capable of performing such a task.

The Area Of Interest (AOI) component was developed specifically for the vehicle scenarios, although it has application to a broader set of problems as well. As introduced earlier in Section 2.1.2, there is a need to provide a data fusion like capability when dealing with large volumes of information, in order to provide other components with for example a distilled summary, or key elements extracted from, a data stream.

The AOI component is used to achieve this in the vehicle simulation scenarios. It subscribes to relevant data stream nodes (e.g. high frequency low level SA streams), but also requests additional information during the initialisation of a scenario, such as traffic light locations and predefined routes. This additional knowledge store is used in conjunction with live data (e.g.

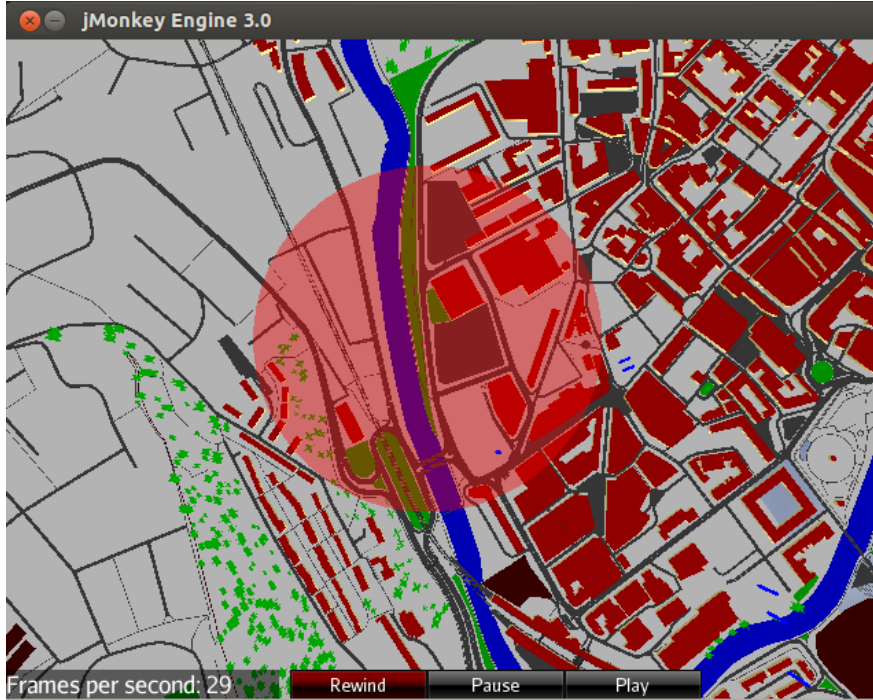


Figure 4-11: Area of Interest displayed in 3D viewer

vehicle positions, traffic light colour states) to provide some specific messages to vehicles (e.g. approaching a red light controlling an upcoming section of route being taken). In the current implementation of the AOI module, constructing this setup requires developer knowledge to configure the initialisation requests, and also to know what relevant information to extract. However, this could be replaced with external configuration files, for example defining RDF request messages and what time they should be issued.

This has the benefit of offloading the computational expense of such calculations from the agent framework to an independent component which can maintain a datastore of potentially useful information. Whilst it could be argued that the agent framework could perform this task, we have adopted this design to increase the application of the framework to a wider set of uses as if data quantity, or message processing complexity increase too far, the slow down to the agent framework could cause unreliable performance (such as cases presented in Section 6.1.2 and 6.1.3 of agents becoming saturated with data). Future work is discussed in more detail in Chapter 7 where the functionality of the AOI module could be replaced with more sophisticated data fusion or stream reasoning if alternative approaches are needed, which demonstrates the ability of the framework to address the problem statements of Section 3.2 concerned with maintaining a generalised distributed systems approach.

The AOI volume itself can be displayed visually in the 3D viewer tool, as shown in Figure 4-11. Here, the volume is represented based on received RDF messages containing the location and radius of the volume, with the radius being computed dynamically, based on the vehicle's speed.



The assumption is that at lower speeds, a vehicle would not be interested in environmental information from far away, whereas at higher speed such information may become important within a few time steps.

We also consider the AOI module as having generalised relevance and capability to aggregate data sources to support the ‘impedance matching’ concept (discussed earlier in Section 3.4.1) as well as potentially supporting the creation of higher levels of SA. In the same way that alternative domains were considered in Section 4.1.3, we propose the use of the AOI module in the context of smart homes, in order to reuse the generalised nature of this capability. As discussed in Section 4.1.1, the BSF was originally conceived of for use in smart homes, and in such a context there are a number of potential entities (e.g. human and animal occupants, plants) combined with overall objectives (e.g. maintain temperature, alert if dangerous gas levels, minimise energy consumption). In this domain, there may be benefits in considering geographical areas of interest such as rooms (to aggregate multiple sensor feeds to some higher level belief such as the room being too warm), or to seek communal benefits where an AOI comprises of a number of flats with access to some shared resource (e.g. hot water reservoir), or a large multi occupancy building considered as an AOI and tracking a higher level metric such occupancy levels during the day. In these scenarios, we see the generic capabilities of the AOI being reused, with its data aggregation capability (i.e. multiple temperature sensors to inform action selection on a room, flat, floor, building level) but also with potential to draw on the SA concept (of exchanging higher level information). Here historical data could be drawn on, to transition lower level perceptions (e.g. lights have been switched off, flat is empty) into higher level SA knowledge (e.g. when this happens on weekdays, flat will be unoccupied until the evening) capable of generating more intelligent responses (e.g. reduce heating, activate cleaning robot). Such capability could assist in the use of institutions in this context, where some limited resource (e.g. available hot water) could be distributed between occupants if a collective approach is adopted (i.e. drawing a limited amount, consuming in serial rather than parallel).

The potential reuse of the AOI capability in other applications, whilst not developed further, is highlighted to support our claims in addressing the problem statements set out in Section 3.2 concerned with a generalised solution, showing application to areas beyond the traffic domain, which is revisited in the following sections discussing a general set of utilities developed for the BSF framework.

#### 4.1.6 RDF Utilities

With the motivation of the problem statements of Section 3.2 to improve the observability of a distributed system, the adopted messaging approach based on the Bath Sensor Framework was presented in Section 4.1.1. The use of this data source to provide a 3D representation of events is discussed in Section 4.1.7, but there is a need for a wider set of utilities to handle these messages, and provide functionality such as logging, replaying, analysis and monitoring. In this section we present the utilities that have been developed to provide this functionality.

```

vin@vin-laptop: ~/BSF/rdfUtils
fileInitializer initialize
[java] INFO: Loading providers for file [classpath:META-INF
ders]
[java] Now connected!
[java] Node creation failed, fetching old one.
[java] Created vehicleSender, now entering its logic!
[java] Now connected!
[java] Node creation failed, fetching old one.
[java] Created simSender, now entering its logic!
[java] log4j:WARN No appenders could be found for logger (
tot.stream.JenaIOEnvironment).
[java] log4j:WARN Please initialize the log4j system prop
[java] log4j:WARN See http://logging.apache.org/log4j/1.2/
for more info.
[java] sent 1 messages per second with 10000000 delay
[java] sent 77 messages per second with 10000000 delay
[java] sent 88 messages per second with 10000000 delay
[java] sent 88 messages per second with 10000000 delay
[java] sent 90 messages per second with 10000000 delay
[java] sent 90 messages per second with 10000000 delay
[java] sent 90 messages per second with 10000000 delay
[java] sent 89 messages per second with 10000000 delay
[java] sent 91 messages per second with 10000000 delay
vin@vin-laptop:~/BSF/rdfUtils$

```

(a) RDF Publisher

```

vin@vin-laptop: ~/BSF/rdfUtils
ders]
[java] connected subscriber
[java] subscribed to jasonSensorVehicles
[java] 0 messages per second
[java] log4j:WARN No appenders could be fo
tot.stream.JenaIOEnvironment).
[java] log4j:WARN Please initialize the lo
[java] log4j:WARN See http://logging.apach
for more info.
[java] 1 messages per second
[java] 0 messages per second
[java] 0 messages per second
[java] 0 messages per second
[java] 0 messages per second
[java] 22 messages per second
[java] 82 messages per second
[java] 89 messages per second
[java] 88 messages per second
[java] 90 messages per second
[java] 90 messages per second
[java] 89 messages per second
[java] 89 messages per second
[java] 91 messages per second
^Cvin@vin-laptop:~/BSF/rdfUtils$

```

(b) RDF Subscriber

Figure 4-12: Example output of rdfTest tool components

## Testing

For a distributed communication framework, a key requirement to support debugging is the ability to investigate low level message exchange. A tool named “rdfDebug” was created, which subscribes to specified nodes and outputs all received messages to the command line (broken down by subject, predicate and object). Whilst this provides some useful information, once message exchange became more complex (with a need to analyse message volumes, rates, filter specific types of messages, etc) then a more sophisticated approach was required.

A key development was the rdfTest package, a command line tool used to assess the performance and reliability of message exchange. This can be started either as a subscriber, in which case the tool outputs how many messages are received each second, or started as a publisher, sending a predefined number of messages each second, or as fast as possible. The premise is straightforward, that for given N messages published each second, there should be N received by the subscriber.

The output of these components is shown in Figure 4-12, where Figure 4-12a shows the publish component producing 91 messages per second, and Figure 4-12b receiving 91 messages per second, demonstrating that message exchange is behaving as expected. Such checks have proved necessary to identify configuration issues and network issues, details of which are presented later in Section 4.2. Also of note is that this tool has been extended to run the same tests but using the JSON message format instead of RDF, results of which are presented in Section 4.2.

Such has been the importance of having reliable message exchange, that a build test (incorporated into the Jenkins<sup>7</sup> continuous integration environment) has been included in the

<sup>7</sup><http://jenkins-ci.org/>, accessed 25th September 2014

BSF package. This calls the `rdfTest` in a non-interactive mode, starting both a publisher and subscriber instance. A predefined number of 6000 messages are published and the subscriber checked that they are all received. Furthermore, a 10 second burst is performed to ensure network reliability, as these messages should all be received by the subscriber. The reasoning is that in the same way as a system would not be extensively used were it known to contain problematic code, use of the BSF framework should be carefully considered if the message exchange performance between components is poor.

## Logger and Replayer

The ability to log RDF data brings about two key benefits: data can be analysed in more detail, and is available for playback. Earlier work developed a logging capability from the BSF connected to Allegrograph (a database tool capable of logging RDF triples as well as supporting SPARQL as a query language).

The logging of data in this fashion builds a repository of simulation data, allowing analysis to be performed on a logged simulation run, exploiting the semantic richness of the RDF data. The query approach was extended further, to provide a replay capability. Here, the replay component is responsible for simulation time management, and at each time step, queries the database for RDF readings taken in that time interval. These are then converted back to BSF data readings, and broadcast to the BSF subscribers. The benefit is that for the rest of the BSF simulation members (e.g. 3D view, RDF monitors) there is no difference between a live simulation, and a replay of a logged experiment. The 3D view was extended with simple control buttons to pause, rewind and play, which publish appropriate RDFs to control the simulation. The replay component will change its mode of operation according to these commands, allowing the 3D GUI to control exploration of the logged data.

The logging and replay capability provides an investigative tool in order to understand why certain events or behaviours occurred for a given experiment. As the 3D view is augmented with agent beliefs (both textual and where appropriate located spatially e.g. route waypoints), institution-issued norms, and simulation metrics, the replay capability can be used to perform analysis of a logged scenario and debug any unexpected behaviour (e.g. identifying the underlying issues in the cases of Section 6.1).

## Monitoring

The final significant component of the developed RDF utility suite is that of realtime monitoring, where measurements of different aspects of system performance are presented to the human system observer (involved in analysis or debugging) as the simulation is taking place. Two main variations have been developed, one analysing low level simulation measures at the network level such as message delays and message quantities, the other variation measuring problem domain specific metrics such as fuel consumption and congestion.

In order to provide such a realtime visualisation of these metrics, a capability is needed where measurements based on received BSF messages can be displayed in a suitable format,

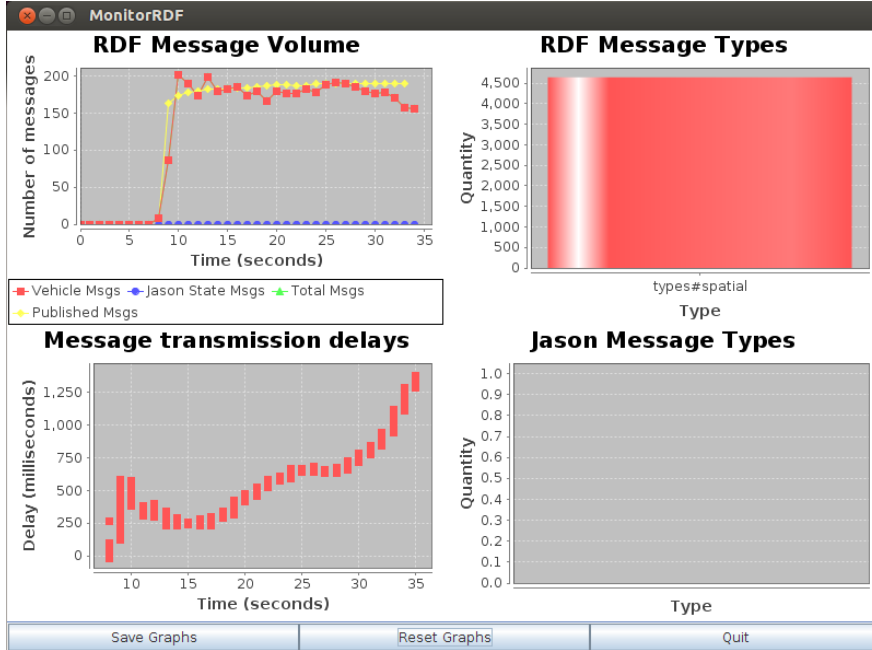


Figure 4-13: rdfMonitor network metrics

which can be updated as the simulation takes place. Whilst the capability of 3D representation described earlier in Section 4.1.7 is suited to displaying spatial representation of events, here we seek a 2D graph representation. To achieve this, we developed a monitoring component using the jFreeChart<sup>8</sup> tool, a Java based framework providing graph capability. As this component receives messages from the BSF, it performs any required metric analysis (e.g. number of messages received per second) and creates a corresponding graph plot point for this result. For post simulation analysis, further capabilities were developed such as saving these data series to csv files, and is used to generate data sets for the various graphs presented in this thesis. The displayed jFreeChart based graphs can also be saved directly as jpeg images, a process which has been used to build up videos based on frames at each timestamp, and these have proved useful to display how a simulation evolves over time.

Specific metrics were developed in this tool to capture aspects of the scenario of interest. This has the benefit of not adding any computational overhead to other BSF components, as the raw data is already published to RDF, the onus of analysis resides with the subscriber.

An example of network specific metrics captured by this tool is shown in Figure 4-13, with four graphs displayed in the tool. The “RDF Message Volume” graph is one of the key measurements, showing the number of RDF messages received each second. When these messages are generated by the rdfTest publisher tool, an additional message is published containing the number of messages sent, allowing two series to be displayed. In Figure 4-13 it can be seen that there is some deviation between the two series, with the number received frequently lower

<sup>8</sup><http://www.jfree.org/jfreechart/>

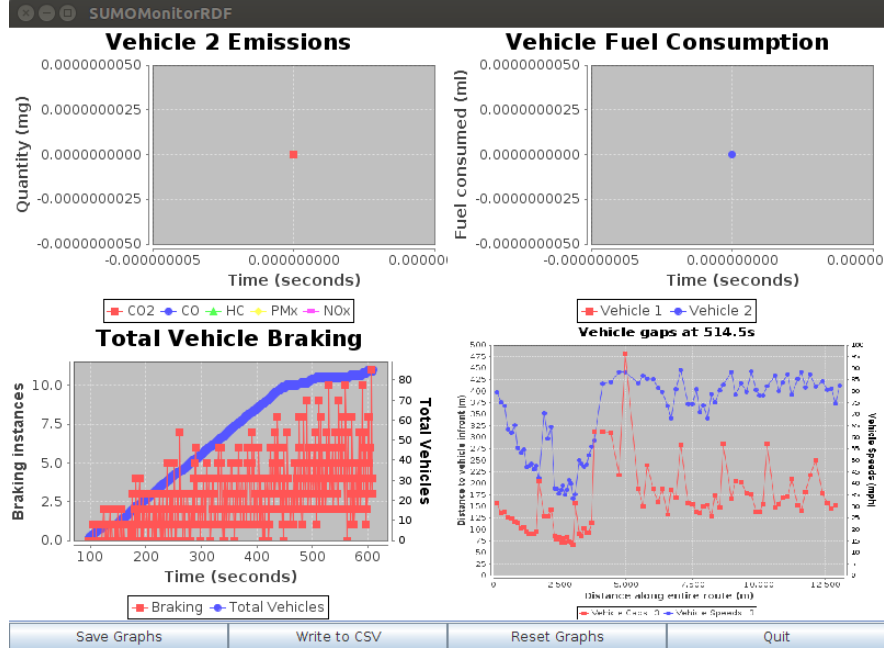


Figure 4-14: rdfMonitor domain metrics

than the number published. When considered in conjunction with the “Message transmission delays” graph shown in the bottom left quadrant of Figure 4-13 which uses the timestamp included in an RDF message, compared to time on arrival at the subscriber, to calculate the transmission time, i.e. the delay from publishing to receiving. In Figure 4-13 it can be seen that over time messages are being received with an increasing delay, and combined with the information from the RDF message volume graph, we can infer that there is a backlog of RDF messages, which is increasing over time. This provides an example of assistance that can be provided to a human attempting to debug the system, by improving their awareness of what is occurring at the network level.

The other two graphs displayed in Figure 4-13 provide information on message types received, in this case only spatial messages have been received, approximately 4500 in total. The Jason message type graph is used in similar fashion when additional logging information is published from Jason regarding types of message exchange between agents (e.g. tell, ask KQML/FIPA messages).

Domain specific metrics were developed for the rdfMonitor tool, with an example of such customisation shown in Figure 4-14. In this example, the most active graphs are the bottom two, firstly “Total Vehicle Braking” which was developed to capture a measure of congestion, on the assumption that along a motorway section, if there is notably more braking in one scenario than another, there may be a higher level of congestion. Additionally shown in this graph is the total number of vehicles in the simulation, as there is a correlation between the number of vehicles, and an overall increase in the amount of braking. This was found to be an unreliable

indication of congestion, so a further measurement was developed, shown in the “Vehicle Gaps” graph. In this graph, each vehicle in the simulation is plotted at its position along the motorway route, with one data point for its current speed, and another for the distance to the vehicle ahead of it. This proved a far more reliable indicator of congestion, as when a cluster of vehicles’ speeds drop, correlated to a decrease in gap to the vehicle ahead, there is found to be some congestion visible in the SUMO GUI. In order to display such congestion behaviour evolving, a video can be constructed based on frames from this graph taken each time step, which provides another means to view the behaviour of the scenario.

These metrics are revisited later in Section 6.2 as part of a more in depth analysis of the results arising from experimentation using the framework.

### 4.1.7 3D Viewer

In order to view vehicle behaviour in the scenarios put forward in Chapter 5, there is a requirement to represent the virtual environment and entities within it, in order to assist the system observer perform debugging of the system. To achieve this, a 3D viewer component has been developed to represent what is occurring in the simulation, including spatial representation of entities in the environment, augmented with other information that is available from the BSF data stream(s). This component also aims to meet the requirement of Section 3.4.5 to provide a means of reviewing results from scenario runs, which was discussed earlier in Section 4.1.3 where the 3D viewer was first established as a separate BSF component. Some inspiration was drawn from developments in the area of augmented reality, where humans have their conventional ‘world view’ augmented with additional overlays of data. In our case, this data is drawn from other BSF modules publishing information, such as agent mind states, institution states, as well as performance metrics of the system itself (e.g. BSF message volume).

The implementation is based on the jMonkeyEngine (Kusterer [2012]) 3D framework, as experience from the TankCoders investigation was that this is a reliable open source platform capable of providing a generic 3D view as required in this work.

This module was developed in two main phases, with the first touched upon earlier in Section 4.1.3 where the 3D view provided by TankCoders was transferred to a new BSF module called “xmppViewer”. This was planned to be a simple receiver of information from the BSF platform, subscribed to vehicle data as well as Jason data streams, and representing them in a 3D world. However, with the logging and replay capability (discussed later in Section 4.1.6) additional GUI controls were added to allow control over the replay of logged data, in order to assist a human observer during the debugging process. These controls can be seen in the earlier Figure 4-5 as Rewind, Pause, and Play buttons at the bottom of the screen, which generate corresponding messages for simulation control to the BSF, which the replay BSF component responds to. This also highlights the increasingly distributed nature of the components, that whilst it is the RDF replayer component which responds to the time control messages in this case, it could be any other component capable of providing simulation time control (e.g. if the simulator was capable of pausing and resuming).



Figure 4-15: Improved 3D xmppViewer tool

To assist the human observer in explaining and debugging the occurrence of (unexpected) behaviours (e.g. those discussed in Section 6.1 of agent failures) the 3D representation shown earlier in Figure 4-5 was expanded to include additional information. We want to provide representation from multiple layers of the simulation (e.g. institution, intelligent agent, network) to assist them in correlating observations (i.e. the vehicle performed an unexpected behaviour) with potential causes (i.e. this happened at the same time that network messages dropped to zero). The result of this is shown in Figure 4-15, where the additional layer of network performance has been added (in this example with message volume spiking at approx 100 messages per second).

We also see this component as offering future capability in support of the problem statement from Section 3.2 regarding observability of the reasoning process and information flow of intelligent systems. Whilst in its current form it presents information to the system debugger from various layers of the simulation, we see potential benefit in using this approach to inform humans of agent beliefs and their reasoning process. This module could also potentially be reused to allow a human to drive a virtual vehicle in order to explore the interaction between human and agent controlled vehicles, which is discussed in the further work of Section 7.2.1.

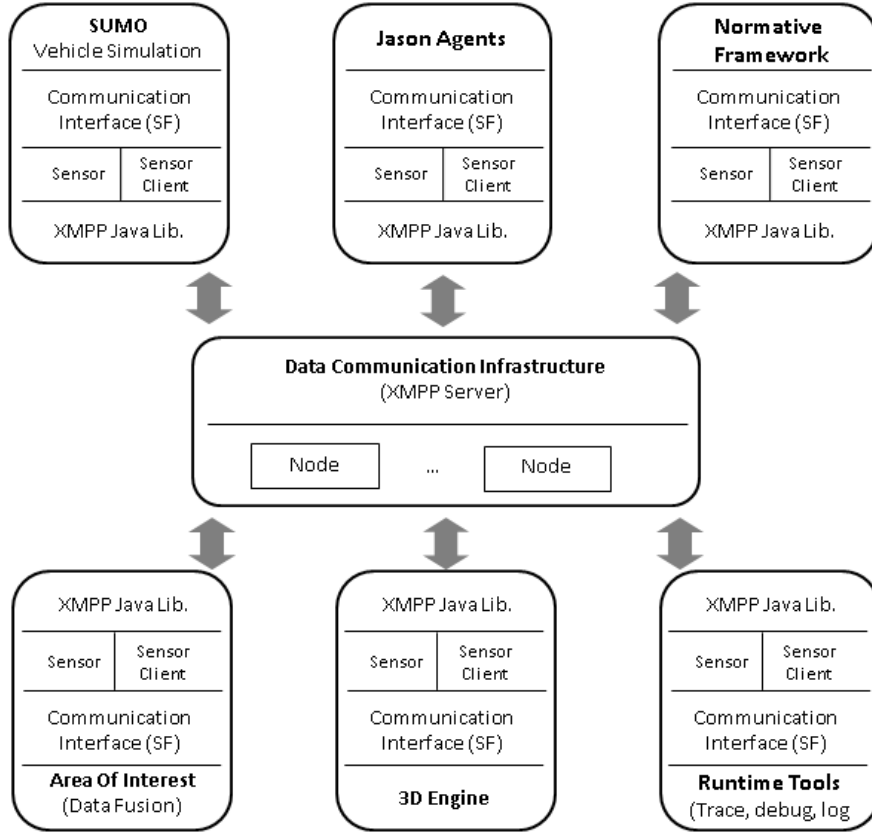


Figure 4-16: BSF module deployment

#### 4.1.8 Final system

A visual representation of the BSF components which have now been introduced, and are used in the remainder of this work is presented in Figure 4-16 (based upon earlier work in Lee et al. [2014]). However, not all of these components are necessarily used at the same time, for example core simulation resides with the SUMO and Jason modules, with the normative framework and AOI modules being used for some scenario variations presented later in Chapter 5. Finally, the runtime tools and 3D engine are largely used as needed by the human system designer and debugger, to view aspects of the simulation.

## 4.2 Performance and testing

For a system based upon distributed components, reliant on timely message exchange between simulation members, performance and reliability are key factors to consider. Validation and testing relating to scenario specifics is discussed later in Section 5.1.3, whilst here we report on findings related to the BSF configuration and achieved performance. This is presented in order to provide a context as to where we found the limits of performance with this system



configuration, in relation to scenarios put forward in Chapter 5 (e.g. could the system of handle higher simulation entity counts).

Some related issues have already been discussed in Section 4.1.3 in both the integration of real platforms (in this case, the arDrone) and simulation performance issues found when using TankCoders and Jason agents. Referring to the earlier results of the arDrone publish-subscribe performance in Figure 4-6 it can be seen that the received rate is always lower than the published rate, a similar situation to that shown in Figure 4-13 where a high volume of message exchange was generated via the rdfTest tool. Similar performance related issues were found when experimenting with TankCoders, which became observable by the reported graphical frames per second rate dropping, correlated with unexpected agent behaviour occurring.

This created the motivation to explore where and why the bottlenecks were occurring, in order to both alleviate the immediate problem and to build confidence and understanding in the overall BSF behaviour. Whilst the distributed nature of the BSF approach made it easy to solve hardware limitations such as CPU demand by compute-intensive 3D software and the Jason platform (by moving them to separate hardware platforms), the network layer proved more challenging.

Initial work focused on the XMPP messaging server component, as any sub-optimal configuration of this module would degrade the entire performance of the framework. The Openfire<sup>9</sup> messaging server was the package first used to provide the XMPP server functionality, and has been found to offer both reliability and good performance. However, when inexplicable performance issues were still present, the decision was taken to seek an alternative to Openfire, at which point ejabberd<sup>10</sup> was deployed. Based on Erlang, there was an expectation this would offer improved performance over Openfire, and in any case would provide a useful comparison. In Figure 4-17 the hardware configuration used to perform some benchmark comparisons is shown, comprising of one desktop PC running the XMPP messaging server under investigation, and another PC running the rdfTest suite publisher and subscriber.

In Figure 4-18, a comparison of these two XMPP servers is shown, where the rdfTest tool has been used to generate a target of 1000 messages per second, with another instance of the rdfTest subscriber recording how many were received. Firstly, it can be seen that for both messaging servers, the achieved published rate is substantially below the target, at approximately 800 messages per second. When running with ejabberd, the subscriber shows a reasonably close performance to that of the publisher, however when used with Openfire the subscriber is receiving approximately 200 messages less than the published quantity each second. From these results, we can draw the conclusion that when using the RDF message format combined with an ejabberd server, we should try to ensure message volume is below 750 per second, otherwise we may hit a performance limitation. There can be less confidence if Openfire is used, and in that case message volume should be kept below 500 messages per second.

However, this did not fully explain the performance issues, as in both configurations there is significant issue with the target publish rate of 1000 messages per second not being achieved.

---

<sup>9</sup><http://www.igniterealtime.org/projects/openfire/>

<sup>10</sup><http://www.igniterealtime.org/projects/openfire/>

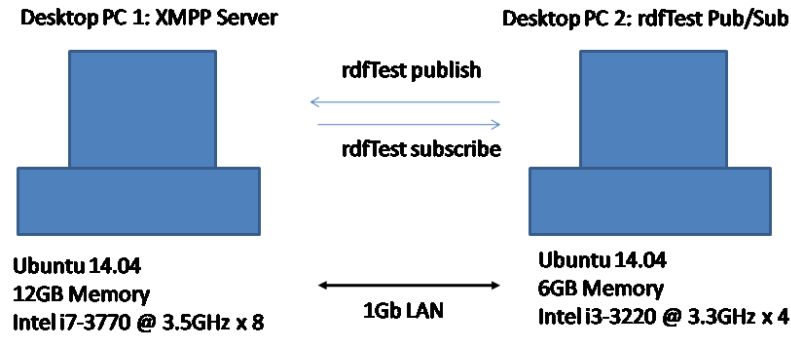


Figure 4-17: Messaging test hardware configuration

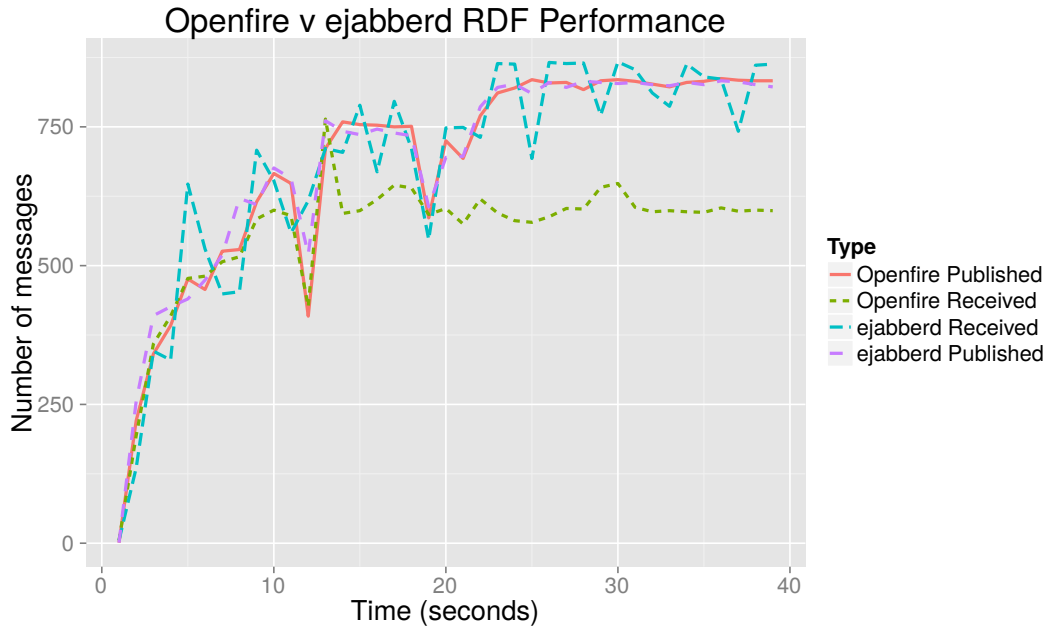


Figure 4-18: Openfire vs ejabberd performance for RDF messages

In other work (Lee et al. [2012]) an alternative to the RDF implementation in the BSF was developed, based on the JavaScript Object Notation (JSON) format. This provides the opportunity to investigate performance differences due to the message format used, covering issues such as serialisation/de-serialisation encountered in generating that message.

In Figure 4-19, the same tests (as used which produced the results in Figure 4-18) were re-run, but this time using the JSON message format instead. In this case, it can be seen that for both Openfire and ejabberd message servers, the publish rate is very close to the target of

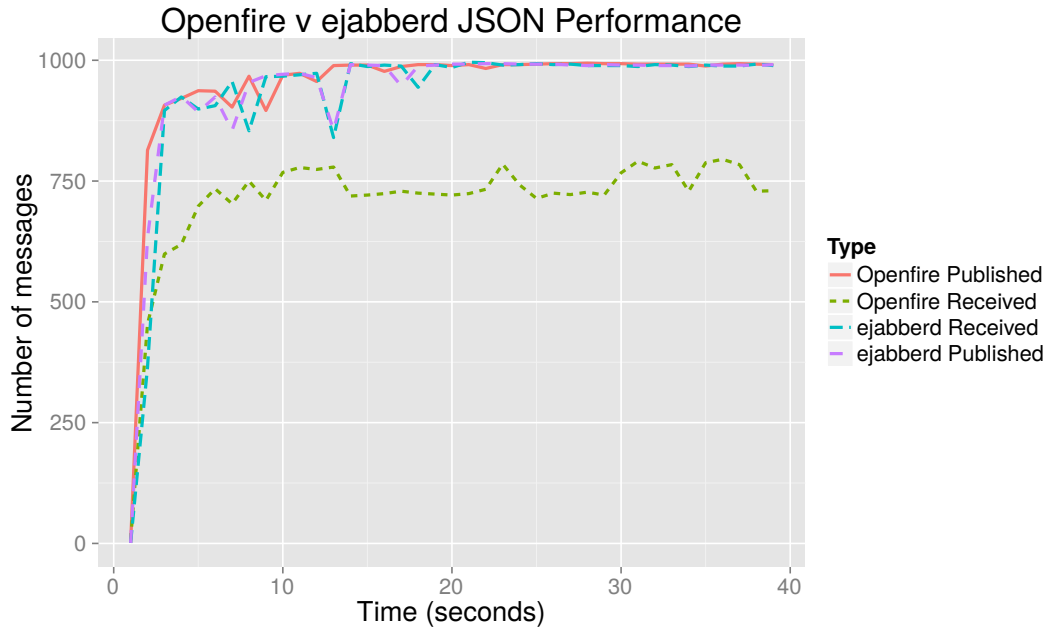


Figure 4-19: Openfire vs ejabberd performance for JSON messages

1000 per second. However, similarly to the findings in Figure 4-18, whilst with ejabberd the subscriber follows very closely to the published quantity, with Openfire there is a significant drop in received messages, approximately 250 in this experiment. From these two experiments, we conclude that JSON messages perform significantly better in the serialise-transmit-de-serialise steps compared to RDF, and that ejabberd provides a more stable messaging server. Of course, effort could be spent in fine tuning parameters, exploring other serializers (which we believe would address the client-side de-serialisation performance issue shown in Figure 4-18), but this establishes our baseline performance expectations of the BSF configuration used in this work.

These proved useful findings, aligned to the general research theme of what to communicate, and how much communication is required. JSON is demonstrated as a more efficient (in terms of message delay and maximum throughput) exchange medium, but lacks the semantic richness available in RDF. By comparison, RDF offers the additional richness, at the cost of performance. The conclusion from this finding, was that if information exchange is likely to be simple (perhaps even predefined) types of messages, at high volume, then JSON may be suitable. For messages where the receiver may not fully understand a simple message (or where additional reasoning about the message may be needed), then providing the volume is not too high, RDF may be a better option. The benefit of the BSF approach, is that both are easily supported, additional formats could be introduced, and that there is a toolset in place (e.g. the rdfMonitor) to assess the performance of the chosen approach.

Related to reliability, the use of Jenkins as a continuous build tool was discussed earlier in Section 4.1.6, improving both the quality of build releases but also including automated tests for

message exchange performance. The SonarQube<sup>11</sup> toolset was investigated as a means to assess the quality of the Java code itself, in order to identify poor practises and code performance issues. This has been incorporated into the core BSF libraries, but has not yet been built into all BSF modules. Practises from agent orientated software engineering were also considered, specifically the approach in SUNIT (Tiryaki et al. [2007]) to developed unit tests for agent behaviour. Future work is considered to incorporate such testing into the agents developed in this work, in order to ensure their expected behaviour as modifications to BSF dependencies may happen over time, and this is discussed in more detail in Chapter 7.

### 4.3 Summary

In this chapter, we have outlined the development process used to construct a system suitable for investigating the problem statements of Section 3.2, operating in the problem domain of intelligent vehicles discussed in Section 3.3, where the requirements outlined in Section 3.4 have been addressed.

We provide details of the Bath Sensor Framework (BSF) architecture as the means of providing our distributed systems capability, and how this supports the aim of observability of data through its use of semantically annotated RDF messages. We then present details of how an intelligent agent framework is integrated with the BSF, with specific agent details used in for experimentation in the domain of intelligent vehicles. Details of the evolutionary process regarding the entity representation of this problem domain are provided, with findings that build confidence in the generalised application of the work, whilst being coupled with a suitable vehicle and traffic simulation package to investigate the chosen problem domain. A number of further components developed to support this work and address the problem statements of Section 3.4 are introduced, including an institution framework which will be used to assist agents in a number of scenarios presented in Chapter 5, a 3D viewer to view scenarios to support analysis and debugging, and an Area of Interest component to provide basic data fusion to the intelligent agents.

Supplementary tools contained in the rdfUtilities packages have also been presented, along with test results establishing the framework is capable of reliably exchanging between 500 to 1000 messages a second depending on configuration choices. This limit provides an upper boundary for scenarios, both in terms of design choices in information exchange, as well as characteristics such as the number of simulated entities which could be reliably handled.

Having detailed the software architecture which is used for the experimentation presented in this work, we now move on to discuss the scenarios which will be implemented on this framework and investigate the problem areas set out in Chapter 3.

---

<sup>11</sup><http://www.sonarqube.org/>, accessed 30th August 2014

## Chapter 5

# Implementation of scenarios

Having outlined the problem statements earlier in Section 3.2, and how these would be considered in the problem domain of intelligent vehicles in Section 3.3, we now present details of the developed scenarios used to investigate these problem areas.

We first revisit the problem specifications discussed in Chapter 3 and present in general terms how scenarios were constructed, based on two contexts of city based and motorway based simulations. We present validation performed of these general contexts, where in contrast to testing and validation discussed in Section 4.2 (which focussed on issues such as network performance), we now consider the integration and correct performance between BSF system components (e.g. alignment of Jason agents, to their SUMO locations, to their 3D viewer representation).

We then present specific scenarios which explore areas of the problem statements set out in Section 3.2, where we focus on scenario details and provide a brief summary of results, with a more in depth analysis of the results from each scenario following in Chapter 6. In Section 5.2 we present two communication approaches to assess if vehicles performing a convoy task can achieve the same performance with a lower level of information exchange. In Section 5.3 this is taken further, where message exchange is performed in a higher SA sense, using route information rather than lower level position details. In Section 5.4 message exchange is considered based on the high level projection element of SA, where vehicle behaviour is modified based on the future state of traffic lights in order to arrive at the light when it is green. These three scenarios provide the SA exploration, and offer insight into the first three problem statements of Section 3.2 regarding building awareness and understanding, observability of the reasoning process and information flow, and finding the balance between quantity and appropriateness of communication.

In Section 5.5 we present a scenario where an intelligent agent has no suitable plan to deal with a socially and contextually complex situation of a driver flashing their lights at the intelligent agent vehicle, where an external institution is used to offer guidance to the intelligent agent. In Section 5.6 a scenario is presented which explores the balance of individual pursuits against greater social welfare, in order to examine the related problem statement from

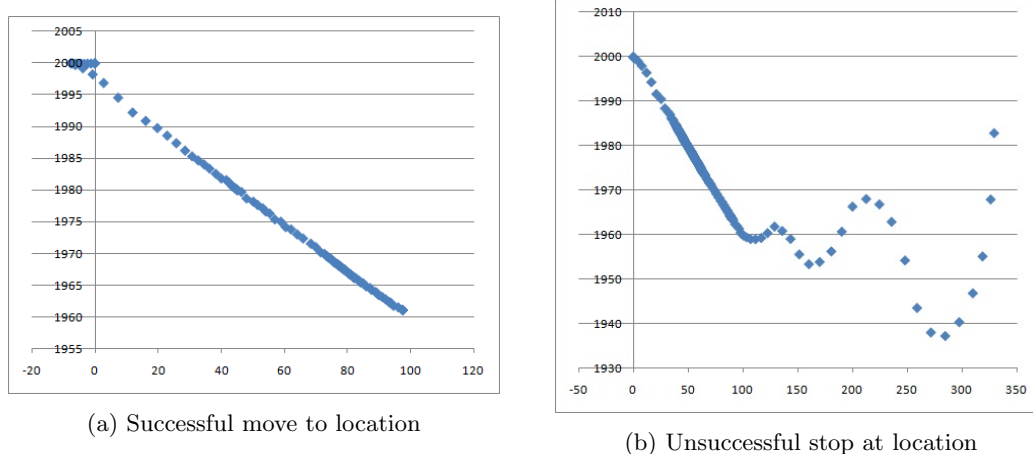


Figure 5-1: Example of early success and failure

Section 3.2, and draws on institutions as a governance structure to regulate agent behaviour. Finally, in Section 5.7 a scenario is presented which deals with the management and coordination of multiple events following a vehicle collision, where the use of multiple interacting institutions is explored as a means of handling the situation.

## 5.1 Building scenarios

The architecture put forward in Chapter 4 is designed to ensure there is general application to other problem domains, assisted by the fact that the BSF framework was originally conceived for and demonstrated in a different domain (of appliance monitoring). Components of the framework have changed over time, showing resilience and that the design remains domain non-specific. In order to assess our proposed design in relation to the problem statements set out in Section 3.2, scenarios are required with the characteristics to explore the problem areas. We first perform an evaluation of the framework in order to assess if it is fit for this purpose, and to establish performance parameters.

### 5.1.1 Developing initial behaviours

The first task explored message exchange between vehicles, where the xmppVehicle approach discussed earlier in Section 4.1.3 could be used to examine cooperative behaviours between vehicles related to the problem statements set out in Section 3.2. Whilst in this scenario there are a number of simplifications made (such as no road following models and no vehicle physics), agent behaviours such as moving to a given position, and communication between these agents could be explored.

In this case, we assume a flat, unlimited terrain, and explore the basic driver agent functionality (as introduced in Section 4.1.2), considering how to move to a given location.

In Figure 5-1 an initial success (a), and a failure (b), can be seen. The xy plot corresponds

to Cartesian points of the simulated vehicle, with the vehicle starting at (2000,0) (in the top right quadrant of the plot) and being tasked to move to (1900,100). At each time step, the vehicle outputs its current location, which results in a plot point. As such, this plot represents the vehicles path, moving from (2000,0) to (1900,100). The difficulty in explaining what this plot represents lead to the realisation that a better way of representing vehicle behaviour was required, and relates to the problem statements of Section 3.2 that there is a need to improve the visibility of agent mind states, in order for humans to understand what is occurring (in this case, to assist in debugging). Regarding performance, in the left Figure 5-1a it can be seen that the vehicle creates a plot point at (1900,100) (i.e. has arrived at its destination), in the right Figure 5-1b it can be seen that unexpected behaviour follows.

In debugging this observed behaviour, a number of general findings were produced. The first finding relates to the difficulty of writing an adequate set of plans, where in this example, some additional plan behaviour was required to slow down the vehicle on approach to the desired location, and then to mark the goal as having been achieved (in order to prevent further plan triggers) when the vehicle has arrived at the intended position (in this case, when the distance remaining to the desired location is below a certain threshold). The second finding relates to intelligent entities not knowing when they are doing the wrong thing, at best resulting in their not achieving a goal and at worst (as in Figure 5-1) making the situation worse (in this case, moving further and further away from the desired location). This provides an example of the type of situation where introducing an external governance structure may help assist agents, and whilst for this specific example the appropriate plan (i.e. to stop correctly at the desired location) should have worked, it is likely that even with an extensive and well tested plan base that not all eventualities would be covered.

At this point, basic variations on the moveTo capability showed expected results, and the system contained sufficient functionality to support the first set of scenarios, as discussed in following Section 5.2 and Section 5.3. However, a more sophisticated domain representation was required, as set out in Section 4.1.3, to provide functionality such as improved vehicle characteristics, road following models, background traffic, in order to explore the rest of the problem statements.

### 5.1.2 City and Motorway

Whilst the previous Section 5.1.1 established basic navigational capability, a more representative and complex environment was desired to support more challenging scenarios, such as navigating around an urban route, rather than straight lines in a virtual desert. With problem statements concerning knowledge and communication, there was a desire to explore how agents could exchange different types of navigational information whilst pursuing some goal. For this reason the urban environment seemed well suited, as routes would increase the complexity of information in the situational awareness sense (e.g. they could be considered as a series of low level waypoints, or a more sophisticated set of directions, through to a single piece of information of the final destination).

In order to support the interoperability of data between system components, there was a desire to build more advanced scenarios from a single source of map data, such that correlating spatial references between all BSF components should be possible. Otherwise, there are pitfalls that different formats may include various assumptions or be built on entirely different coordinate reference systems (e.g. geodetic versus geocentric). Whilst BSF modules may wish (or need to) convert to their own reference system, if data can be referred back to a single originating source, then it is assumed some data conversion (e.g. of coordinates) can take place.

With this motivation in mind, the OpenStreetMap (OSM) format (as introduced earlier in Section 4.1.3) was chosen as the reference map data for these scenarios, due to its wide spread support in applications, its open availability, and ease of access to the data content (as it is based on XML representation). A tool (osm2world, Knerr [2013]) was found to convert this data to a 3D model (as discussed in Section 4.1.7), and combining these capabilities gives accessible map data which can be processed by our agents, and the ability to view the agents movements in a 3D landscape. The expectation was that when viewed in the 3D xmpViewer tool, this would provide a more intuitive view to support debugging of agent behaviour, than the 2D representation shown in Figure 5-1.

We elected to create the scenarios from source (versus attempting to use pre-existing models, maps, and routes) for a number of reasons. The key motivation was to develop a comprehensive understanding of the process so that if something goes wrong or unexpected results occur, we have full access to, and knowledge of, the whole data set. In other words, there is no black box of proprietary models or data somewhere which we have no control over. Such an approach also increases the awareness of assumptions made, so that when the results are being analysed these can be considered in (for example, whether height data has been included in the model, as that may have an impact on vehicle behaviour). There are also additional benefits, as by selecting to use open source data we are able to distributed our models and processes used, and interact with the wider community using the real data when issues are encountered.

The selection of these two contexts also benefits the experimental approach in terms of testing assumptions which could be unconsciously included in any of the scenarios. For example, if only a city context is used, then vehicles speeds would be unlikely to exceed 30mph and there would be no multiple lanes to reason about. Similarly, if only the motorway context were used, then there would be no reasoning about city artefacts such as traffic lights and pedestrian crossings, and routes may be significantly simpler.

### 5.1.3 Validation and testing

In this section we present the validation and testing of the scenario content, primarily whether location information is accurate, as this is key to the whole scenario, both in terms of agents moving to sensible locations, and assessing their performance (e.g. route deviation). This differs to the testing described in the previous Section 4.2 which looked at issues around underlying framework performance, instead we consider the area of domain performance.

As introduced earlier in Section 5.1.2, investigations shifted from a flat desert like terrain,



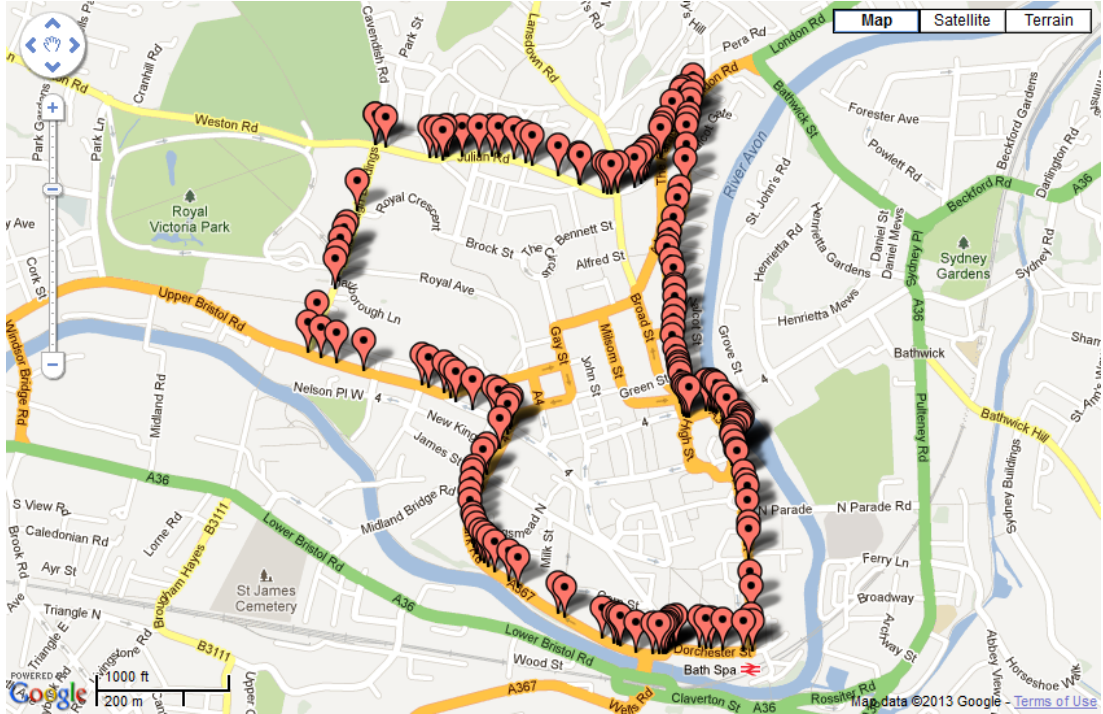


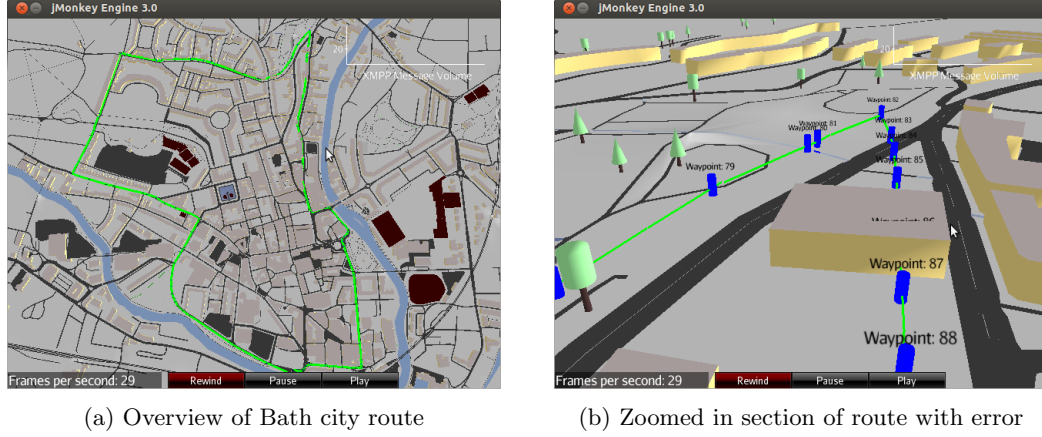
Figure 5-2: Route waypoint verification (Map data ©2013 Google)

to a city context. Initial work explored using the OSM map data and basic route finding capability, to generate a set of waypoints representing a route around a city centre. In this case, the city of Bath was used, and the generated waypoints exported as Google Map points (in KML format), which could then be overlaid onto a Google Map in order to visually check the output.

The results of this can be seen in Figure 5-2, showing visually well aligned points along a road segment around the periphery of Bath city center. By generating these waypoints from the OSM map data, but showing them against Google map data, we have validated two steps: that the OSM map data we are using is accurate (as compared to Google's), and that the process of extracting the locations of waypoints along a calculated route is functioning as expected.

Following this, we move on to validate the 3D model created of the city center using the `osm2world` (Knerer [2013]) package. The motivation for this validation is that we wish to use the 3D component as a means to view what an agent is doing, as well as what it is thinking, in terms of representing beliefs and goals visually. A specific example of this is creating a simple spatial model in the 3D view for an agents desired location (e.g. a waypoint). If the models are not calibrated and validated, then there is uncertainty whether an object appearing visually out of place is due to some issue in the agents calculation, or inaccuracy in the 3D representation.

Prior to the integration of a more advanced vehicle and traffic simulator there was no road following logic, and so vehicles would simply follow the set of waypoints received. Therefore,



(a) Overview of Bath city route

(b) Zoomed in section of route with error

Figure 5-3: Route alignment problem shown using xmppViewer component

since the results of Figure 5-2 suggest the waypoints are accurate, if we trace a vehicles route as it moves between these waypoints, and overlay this into the 3D representation, we can assess the alignment between the model and route. This creates a significant test of the BSF spatial data itself, as whilst the previous test provided a comparison of waypoint data, in this test we are representing a received vehicle BSF message, extracting the spatial location, and representing this in the 3D world.

We use the 3D component of the framework (as described in Section 4.1.7) to highlight this issue, where in Figure 5-3a the route taken by a vehicle is seen overlayed on the 3D city model, and it can be seen there is an issue between this initial alignment. Whilst the route appears correct in some respects it is also suffering problems with accuracy, most visible in the top right section of the route, as shown more clearly in Figure 5-3b. To highlight the issue, in Figure 5-3b the agent's waypoints are shown as blue spheres, with corresponding names (e.g. 'Waypoint 88' at the bottom of Figure 5-3b). In this case, it can be seen that the vehicle is moving between the waypoints as expected, which follows the general shape of the road, but seemingly shifted to the left of the 3D model road.

The issue was found to be due to incorrect coordinate conversion in osm2world, and with this corrected and the 3D model regenerated the problem was resolved. However, this does highlight a wider issue, that in constructing distributed components there is a need to ensure their data sets are aligned, whether this be in units used in messages (where the semantic annotation may assist, if unit types used are specified in a delivered message) or in visual representations used.

## 5.2 Perception driven scenarios

The first scenario investigates the problem area of information and knowledge exchange, in a domain context based on a convoy of vehicles. The topic of vehicles forming 'platoons' or convoys was introduced earlier in Section 2.3.2, and we focus upon this as a relevant domain

development, where the implementation of such capability in the real world is still under investigation.

In this scenario we investigate the problem statement from Section 3.2 of finding the balance between quantity and appropriateness, and assess whether it is appropriate and beneficial to communicate as much as possible (at high frequency, low level SA data) or whether the same domain performance can be achieved when exchanging less data between entities. We represent two options: *data-push* versus *data-pull* strategies. In the push case, vehicles would broadcast their data i.e. rather than generated by a request, they simply output position information at each simulation step. By comparison, the pull scenario is based on vehicles requesting data from other vehicles, at a time of their choosing, in order to manage and possibly reduce their communication.

From a domain perspective, the data push adopts an approach to convoy management based on the assumption of high communication between vehicles, compared to the pull approach which should result in less communication and demands on vehicle to vehicle infrastructure, thus reducing network and computational load, but at the risk of reduced domain performance.

We focus on convoy maintenance here, and do not consider convoy creation or dissolution, and start from a position where three vehicles are following a convoy lead vehicle. Part of this ‘convoy-follow’ behaviour involves determining whether the convoy is heading (at least partly) in the desired direction of the new convoy joiner. On joining the convoy, there is an abandonment of route-planning responsibility as part of the ceding of individual autonomy to the collective convoy, and instead navigation becomes the task of following the trail of the vehicle in front.

Both scenarios have the lead vehicle following the same route, from the lat/lon coordinates of 51.377831,-2.360473 to 51.381086,-2.366266, a distance of approximately 650m. This route is shown in Figure 5-4 using the 3D component view. The route is highlighted by the viewer, but the image has also been annotated to show the start and end points.

### Data push strategy

The variation of this scenario based on the approach of data push implements the following approach:

1. Only the lead vehicle knows the final destination and the route from the start position to the end location.
2. The lead vehicle’s coordinator agent starts the movement by sending a message to its driver agent to move to the predefined position:  
`send(driverAgent, achieve, calcStandardRoute)`
3. The driver calls the external route calculation package, calculating a route from the lat/lon of 51.377831,-2.360473 to 51.381086,-2.366266. The driver agent is returned a set of waypoints to pass through.

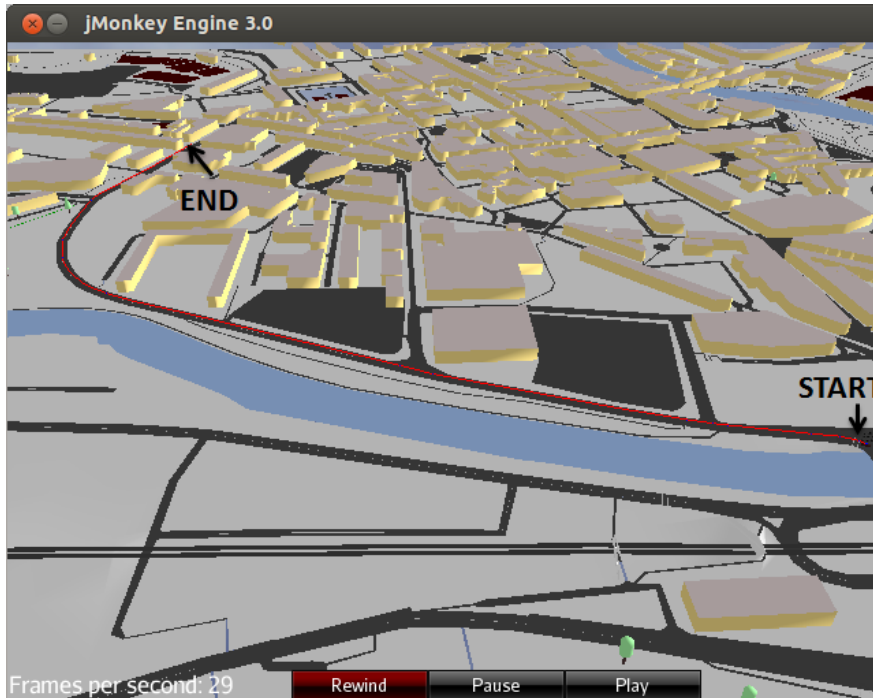


Figure 5-4: Vehicle convoy route around Bath city centre

4. The coordinator agent asks the driver agent to achieve a movement through these way-points:  

```
send(driverAgent, achieve, followWayPoints).
```
5. Each vehicle in the convoy starts a convoy member agent.
6. Each convoy member agent is told the vehicle's driver agent name (in order to be able to send messages regarding updated positions to move to) and the name of the convoy member agent of the vehicle behind (in order to push data to the correct vehicle).
7. On every simulation update cycle, the coordinator agent tells its convoy member agent and driver agent the vehicle's new position.
8. When a driver agent receives a position update, it uses this to calculate the distance remaining to the `desiredXZ` and perform any necessary actions (e.g. course corrections, or stop if at that location).
9. When a convoy member agent receives a position update from its coordinator agent, it performs a data push to tell the following convoy member agent this new position.
10. When a convoy member agent receives a position updates from the vehicle ahead, it tells its driver agent as a new `desiredXZ` followed by the request to achieve `moveToKnownPosition`.

### Data pull strategy

The variation of this scenario based on the approach of data pull implements the following approach:

1. Only the lead vehicle knows final destination and the route from the start position to the end location.
2. The lead vehicle's coordinator agent starts the movement by sending a message to its driver agent to move to the predefined position:  
`send(driverAgent, achieve, calcStandardRoute)`
3. The driver calls the external route calculation package, calculating a route from the lat/lon of 51.377831,-2.360473 to 51.381086,-2.366266. The driver agent is returned a set of waypoints to pass through.
4. The coordinator agent asks the driver agent to achieve a movement through these waypoints:  
`send(driverAgent, achieve, followWayPoints).`
5. Each vehicle in the convoy starts a convoy member agent.
6. Each convoy member agent is told the vehicle's driver agent name (in order to be able to send messages regarding updated positions to move to) and the name of the convoy member agent for the vehicle ahead (in order to pull data from the correct vehicle).
7. Each convoy member agent starts a convoy management plan, in order to handle the data pull aspect. At a parameter-specified frequency this plan uses the KQML performative `askOne` to ask the vehicle ahead for its current position.
8. When a convoy member receives a reply containing the position of the vehicle ahead, it sends this to its driver agent as a new `desiredXZ` followed by the request to achieve `moveToKnownPosition`.

The hypothesis of this scenario is that the data push approach will perform well in the task of convoy movement, with the metrics expected to show close cohesion between the convoy vehicles. For the variation based on a data pull strategy, the expectation is that there will be a decrease in message volume but that there may be some trade off in reduced convoy performance (e.g. the vehicles may deviate from the lead vehicles' route, or that they may become separated).

The results of this scenario are presented in Section 6.3.1, where in summary we find a significant drop in message volume when adopting the data pull approach, whilst domain performance in convoy behaviour remains consistent in both scenario variations.

### 5.3 Waypoint exchange

In the previous scenario based on perception exchange, the information exchanged could be considered in the Situational Awareness sense of being low level, with agents exchanging information at the perception level of SA rather than comprehension or projection. In this scenario, we extend the previous perception exchange approach by communicating a higher level of information. In essence, we seek to explore shifting from high frequency ‘vehicle N is at position X,Y,Z’ to lower frequency ‘vehicle N is moving to this position via these waypoints’.

Initially a high level approach was considered, to exchange ‘vehicle N is going to this position’ and allowing the agents to calculate the corresponding route. A route planning and navigation library<sup>1</sup> was integrated to provide agents with the capability to request a route to a given (lat-lon) location. However, it was found that alternative routes could be generated (for example, vehicles further behind the lead vehicle may find a quicker or shorter alternative route). Furthermore, this seems to break away from the notion of a convoy; if each vehicle is calculating a route from the lead vehicles given destination, then the behaviour is more like vehicles which just happen to be going in the same direction than a collaborative convoy.

The question of how much convoy-like behaviour is adopted can also be raised in this approach based on exchanging waypoints. There may be a need for additional functionality and interaction, where if the lead vehicle deviates, a new set of waypoints is sent to the convoy, and that the convoy members are checking that they are still following the leader. However, in the implementation of this scenario, we stop short of implementing such functionality, as there is a risk of becoming drawn in to implementing increasingly sophisticated domain specific solutions, demonstrating domain performance but not directly adding to the research questions being addressed. In this case, we are interested in communicating less information, whilst supplying agents with improved knowledge and understanding to achieve their goal. For the domain of vehicles, the exchange of waypoints is demonstrating this as this is considered a higher level of knowledge exchange, and offers agents a richer data set to reason about as they now know the entire route plan rather than a given x,y position to move to as in the perception based scenario.

The waypoint exchange scenario implements the following approach:

1. Only the lead vehicle knows final destination and the route from the start position to the end location.
2. The lead vehicle’s coordinator agent starts the movement by sending a message to its driver agent to move to the predefined position:  
`send(driverAgent, achieve, loadBathRoute)`
3. The driver loads a predefined set of waypoints, based on a route starting from the lat/lon of 51.377831,-2.360473, performing a route around the city, and returning to the same location. The driver agent is returned a set of waypoints to pass through.

---

<sup>1</sup><http://sourceforge.net/projects/travelingsales/>, accessed 30th August 2014

4. The coordinator agent asks the driver agent to achieve a movement through these waypoints:  
`send(driverAgent, achieve, followWayPoints).`
5. Each vehicle in the convoy starts a convoy member agent.
6. The coordinator agent asks the driver agent for the set of waypoints, and then tells the convoy agent this list.
7. The coordinator agent then tells the convoy agent the `useWaypoints` belief, so that the same set of plans can be used as in Section 5.2 but without updating `moveToKnownPosition` (as this is specified in waypoints rather than vehicle ahead position).
8. The coordinator agent informs the convoy agent of the vehicle next behind in the convoy, and is then asked to achieve `passBackWaypoints`.
9. This causes the convoy agent to pass back the waypoints to the given vehicle, and it then asks that vehicle's convoy agent to achieve `followWaypoints`.
10. In the driver agent, the `followWaypoints` plans work through the list of waypoints, setting the `desiredXZ` location to the current waypoint and moving onto the next waypoint once the current waypoint has been reached.
11. When there are no more waypoints remaining, the vehicle has arrived at its destination.

The hypothesis of this scenario is that we should see similar behaviour to the scenarios based on perception exchange in Section 5.2, but with reduced dependency on high volume communication. There should also be observable domain performance improvement, through a reduction in reactionary braking and acceleration, due to removing the dependency on moving to the last position of the vehicle ahead. Although not directly explored in this scenario, the availability of such richer information would also allow the agent improved reasoning (i.e. remaining trip time).

The results are presented in detail in Section 6.3.2, where in summary we find similar levels of communication to the data pull scenario discussed earlier in Section 5.2, but improved domain performance in a reduction of acceleration and braking events (i.e. smoother driving behaviour).

## 5.4 Projection of future events

In this scenario, the high level projection component of Situational Awareness is examined, where the future state of some perception would be used to improve an agents understanding of what is likely to happen. To illustrate the benefits such SA could have, we construct a scenario in which this projection component is used to generate alternative behaviours (compared to such projected states not being used).



For the problem domain of intelligent vehicles, we select traffic lights as a mechanism to demonstrate such behaviour. Reviewing these from a SA perspective, we can identify three levels and varying sophistication of the potential response:

1. *Perception*: A specific feature of the traffic light, or the traffic light itself, is detected. For example, that the traffic light is red.
2. *Comprehension*: Combining a perception with domain knowledge could result in a higher level comprehension about the traffic light, for example, the traffic light is red so I have to stop.
3. *Projection*: A higher level combination of previous comprehensions, additional perceptions, and temporal features, to result in a high level awareness, projecting what future events may occur. For example, the traffic light turned red, but I'm 100m away and by the time I arrive it will be green, therefore do not slow down significantly.

The notion of planning around future traffic light states was introduced earlier in Section 2.3.2, with a number of research efforts under way to, in effect, improve the coordination between vehicles and traffic lights, such that vehicles arrive at lights when they are green in order to improve fuel consumption and reduce emissions. We draw on the approach discussed in Volkswagen [2013] where vehicles are fitted with a device which indicates to the driver what speed to travel at (below the speed limit), in order to arrive at the next traffic light when it is green.

Whilst this forms the general context for this scenario, the work in this thesis considers autonomous agents controlling the vehicle, rather than a human driver. In this case, we are now faced with one of the problem statements put forward earlier in Section 3.2 of how an agents' autonomy can be managed. When a human operator is presented with such driver aide information, they are given the choice of whether to follow this suggestion or not, and we wish to replicate such a system for our autonomous agents.

It is for this reason that we introduce the institution framework set out in Section 4.1.4. We wish to offer the agent guidance for what to do in this situation, with the expectation that this will be followed, in a somewhat more mandatory fashion than the human driver-aide alternative so as to curtail the agents autonomy over this specific decision. The motivation is that the agent may have a conflicting goal (e.g. get to destination as quickly as possible) which may override the suggestion to slow down. This has two benefits, firstly in simplifying the decision-making structure required for the agent (by removing complex conditional if-then-else statements), and secondly in allowing external controlling or coordinating entities to assert direct control over vehicles when required.

The route is shown in Figure 5-5, where the SUMO GUI is used to provide a view of the map, which has then been annotated with white arrows to indicate the direction of the route, starting from the indicated "START" position through to the "END" location. The numeric annotations 1, 2, and 3 are used to denote the location of traffic lights which control junctions along the vehicle's route.



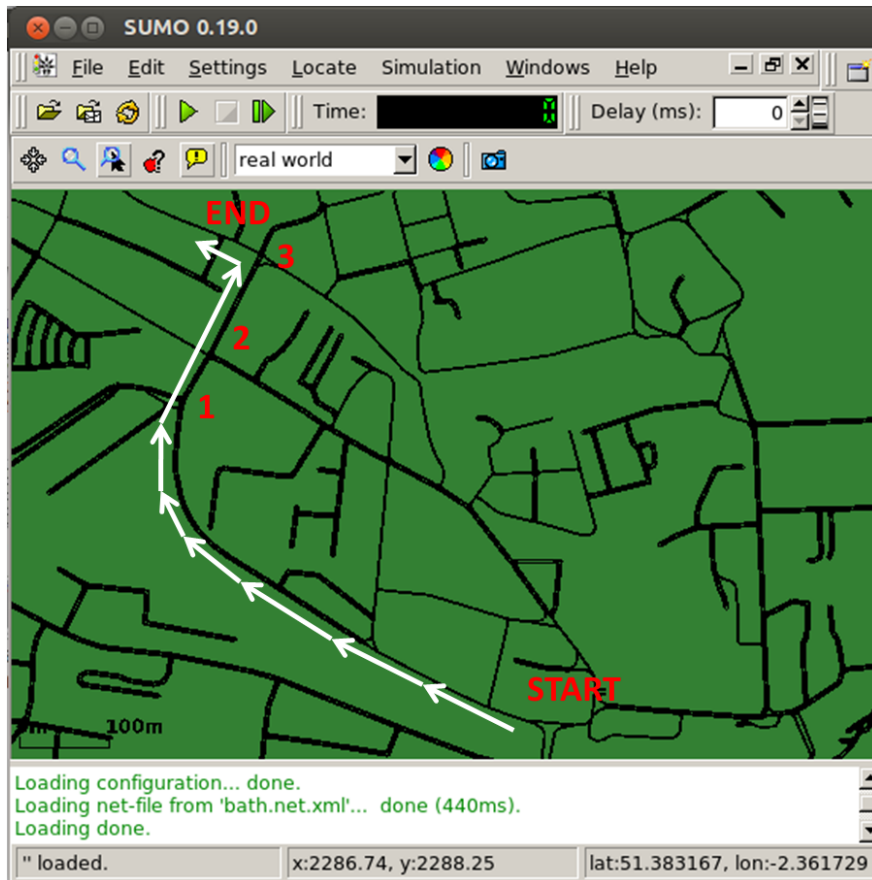


Figure 5-5: Vehicle route in Bath city centre

In this scenario, we need to provide members of the BSF framework with additional information not at the level of immediately extractable data (e.g. position, speed, low level SA type) already being published by the simulator, but at a higher level, with additional data processing to transform information available from the simulator into higher SA information. For this task, we draw on the Area of Interest (AOI) module (presented earlier in Section 4.1.5) which provides the following functionality for this scenario:

1. On startup, the AOI module queries the SUMO simulation to build up a list of traffic lights, their position and what road lanes they control.
2. Jason controlled vehicle `jasonCar1` enters the simulation.
3. The AOI module receives the creation RDF message, and registers `jasonCar1` as a vehicle requiring an area of interest volume.
4. The AOI module queries `jasonCar1` route, and identifies traffic lights which govern any part of that route.
5. The AOI module receives `jasonCar1` position and speed updates, and calculates an AOI volume based on this.
6. If the AOI module calculates an upcoming traffic light inside the AOI volume, it publishes an `upcomingLight` RDF message containing the distance to that light, and its current colour state.

With this process in place, we have a general solution to detect if there are upcoming traffic lights along a vehicles route, with the computation and alerting offloaded to the AOI module. However, there is a question of who should act on the AOI `upcomingLight` message. The vehicle agent could handle this, for example performing some analysis of distance to the light, and modifying its speed accordingly. But, as was just discussed, here we are interested in this situation being coordinated and resolved via an institution.

In this case, we consider two possible variations. The scenario can be run without an institution, in which case no entity processes the AOI generated message of `upcomingLight` and the vehicle continues as it would normally. Alternatively, we introduce the institution shown in Listing 5.1 to provide additional governance in this scenario.

In the institution definition shown in Listing 5.1 we highlight the key component, in the rule definitions where an `upcomingLight` event generates the institutional event `iniOblSlowDown(Agent)`, resulting in the obligation `obl(reduceSpeed(Agent))` being issued. This results in the the following scenario taking place:

1. The institution receives the `upcomingLight` message, and if the distance is between 100m to 300m, while the light state is red, the institution is triggered with the event `upcomingRedLight(jasonCar1)`.
2. The institution resolves this external event and issues a `reduceSpeed(jasonCar1)` obligation, which is published as a BSF message.

### Listing 5.1: Traffic lights institution definition

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Traffic lights institution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% institution trafficlights
institution trafficlights;

%% component
type Agent;

%% exogenous event
exogenous event deadline;
exogenous event reduceSpeed(Agent);
exogenous event upcomingRedLight(Agent);

%% obligations
obligation fluent obl(reduceSpeed(Agent), deadline, vioSlowDown(Agent));

%% institutional events
inst event iniOblSlowDown(Agent);

%% violation event
violation event vioSlowDown(Agent);

%% rules %%
upcomingRedLight(Agent) generates iniOblSlowDown(Agent);
iniOblSlowDown(Agent) initiates perm(reduceSpeed(Agent)), obl(reduceSpeed(Agent),
    deadline, vioSlowDown(Agent));

%% initiate permissions and empowerment
initially perm(deadline);
initially perm(iniOblSlowDown(Agent)), pow(iniOblSlowDown(Agent));
```

---

3. The Jason agent controlling `jasonCar1` receives this message, which creates a `reduceSpeed` belief.
4. The `reduceSpeed` belief causes the agent to initiate a `shortCruise` plan, which sets a slow speed (7m/s) for 35 seconds.
5. After 35 seconds, the speed control is reset to automatic, and normal vehicle behaviour resumes.

With this scenario outline, it can be seen that the approach lacks some sophistication, taking a somewhat crude approach to both distance to the light, and what speed to select (for example, this could be improved by calculations based on distance remaining to light, predicted time of

colour change to green, etc). However, we wish to avoid becoming focussed on domain specific improvements and tuning, and instead to demonstrate the effect and feasibility of a generalised approach in the use of institutions to this domain.

The hypothesis of this scenario is that when issued with an appropriate speed reduction obligation, the longer time taken to arrive at the traffic light which would have been red, has meant that it has changed to green. As such, the expectation is that with the institution active the vehicle should experience less stop-start behaviour, which could result in reduced fuel consumption and emissions.

The results of this scenario are presented in Section 6.3.3, where in summary we find the institution involvement results in the vehicle arriving at the traffic lights when they are green, reducing the amount of braking and acceleration required which improves the fuel consumption of the vehicle.

## 5.5 Resolving problematic situations

In this scenario we focus on the problem statement from Section 3.2 regarding problematic situations, by which we mean cases where an agent may not know what to do (i.e. no appropriate plan for a given set of events), or may not be aware it should take some action. We highlight two facets of the problem, firstly the difficulty in providing all potentially required knowledge to an agent prior to runtime, and secondly the difficulty that arises when interacting with humans due to the ambiguity in interpreting signals and intent from the human driver.

This ambiguity of interacting with humans links to a problem posed in Chapter 3 regarding the introduction of new technology, where in a case similar to the example from Section 3.1.2 of how new motor vehicles should be regulated (at that time dealing with horse transport and human pedestrians), there is a generalised issue of how new technology can be introduced to an existing domain. For the problem domain of intelligent vehicles, it seems likely that there will be human controlled vehicles present on roads for the considerable future, and unless dedicated autonomous-vehicles-only lanes are established (e.g. as done with the autonomous transport system at Heathrow airport discussed in Section 2.3.1) then these intelligent vehicles will have to handle interacting with human drivers.

With this motivation in mind, we put forward a scenario based in a motorway context, where one vehicle is being slowed by another vehicle and wishes to get past. One option for the faster vehicle would be to change lanes and overtake, however in this scenario we explore the indication of the wish to overtake, manifested through flashing of front headlights. In the UK, a document referred to as the Highway Code<sup>2</sup> provides road users with both rules and advisory information for general road use. Regarding the use of flashing headlights, it states: “Only flash your headlights to let other road users know that you are there. Do not flash your headlights to convey any other message or intimidate other road users”. Given this knowledge, autonomous vehicle agents would be expected to pay no heed to the vehicle wishing to get past. However, considering this as a coordination problem, it may be desirable to allow faster moving

---

<sup>2</sup><https://www.gov.uk/highway-code>, accessed 30th August 2014

traffic past, to reduce multiple lane congestion, as well as to provide the intelligent agent with an (external) capability to resolve the meaning.

Further more, there may be cases where a vehicle wishes to get past another vehicle where they do have the right to enforce this ‘move out of the way’ behaviour, for example in the case of emergency vehicles responding to an incident. Whilst vehicles have an obligation to move out of the way, there is some ambiguity in how they should achieve it, for example pulling in front of another vehicle in an different lane, speeding up to get out of the way, and so on. As developments such as vehicle to vehicle communication (as discussed earlier in Section 2.3.2) begin to appear, it creates the opportunity to improve such coordination, for example as explored in Bilstrup et al. [2007] where V2V messaging is used to coordinate clearing a path for emergency vehicles. Work has also taken place in the SUMO traffic simulator being used in this thesis, where Düring et al. [2014] explore adapting traffic light and vehicle responses to the requirement of emergency vehicles to get past quickly and efficiently.

In cases of understanding another vehicle’s signal (in this case, of desire to get past), there is difficulty in resolving what the signal means (flashing of lights, blue lights to indicate move out of the way) whether a move is mandatory, and how to implement such a move (e.g. coordinating a move into another lane). This can be further complicated by the context, for example, in the UK when seeking to pull out at a junction, vehicles sometimes flash their lights to indicate they are allowing you to pull out ahead of them, so in some situations a flash of lights can mean something very different. This also relates to the knowledge and understanding SA theme, as vehicles try to understand what is happening based on ambiguous low level percept of a light flash, instead of more developed higher level information.

For this scenario context, we explore the use of an institution to resolve such ambiguity, and issue appropriate obligations to vehicles as required. This creates the possibility for the institution to enforce required moves (such as to allow emergency vehicles past) but also to take a more society-centric view, where the desire of one vehicle (to remain in its current lane) may be outweighed by the desires of other vehicles (to continue in that lane at their current speed), allowing institutions to manage traffic flow.

Similarly to the previous scenario in Section 5.4, the Area Of Interest (AOI) module is used to inform Jason controlled vehicles if there is an upcoming vehicle detected in the AOI volume, for which the general process is:

1. On receiving vehicle related BSF messages, the AOI updates a list held of vehicle geospatial data.
2. On receiving vehicle creation messages sent from the Jason agent, the AOI module updates its list of Jason controlled vehicles.
3. After receiving new vehicle BSF messages, the AOI module iterates through its list of Jason vehicles, checking if any vehicles are inside its AOI volume.
4. If a Jason vehicle is detected inside the vehicles’ AOI volume, then a BSF message is generated, containing the detected vehicles name, location and if both vehicles are in the

same lane.

This generates the functionality for Jason vehicles to be informed about vehicle detections within their area of interest (and could be easily extended to provide a generic detection capability). However, the final reasoning about this detection is handled by the Jason agent itself, using the following process:

1. The Jason BSF subscriber receives generated AOI messages published by the AOI module, and extracts detected vehicle name, position, and whether it is in the same lane.
2. For detected vehicles in the same lane, a percept is generated of the form `detectedVehicles (Name, X, Y)` and added to the Jason agent for which the detection is intended.
3. On receiving a `detectedVehicles (Name, X, Y)` belief update, the Jason agent calls a method `checkCollisionVolume (Name, X, Y)`.
4. This method is coded in a Java extension provided to the vehicle agent, where a higher resolution check is performed to assess if this is in the vehicles predicted collision zone (a small area immediately ahead of the vehicle, of the space it will occupy in the next few seconds).
5. If the detected vehicle is within this volume, and this is the first time it has been detected, a belief `detectionInCollisionZone (Name, Distance)` is added to the vehicle.
6. If all previously detected vehicles in the collision volume have now left it, a belief `collisionVolClear` is added to the agent.
7. For detected vehicles, if they are between 46m to 65m away, the vehicle flashes its lights. If they are less than 46m, the vehicle brakes hard to avoid a collision.
8. On receiving a `collisionVolClear` belief, the vehicle resumes its previous speed.

This explains the general scenario, and it can be run at this stage to see the desired effect; a faster vehicle would approach a slower vehicle, flash its lights, and eventually have to brake hard to avoid a collision. However, this is where we draw on institutions to offer a solution to the problem statement of Section 3.2 concerned with resolving problematic situations. The slower moving vehicle may have perceived the flash of lights, but not contain appropriate reasoning to decide what this means. According to the rules for the situation, no action is required, and instead we are dealing with a social convention (or, inferring meaning when there is no clear process to do so). It is for this type of situation where we consider the use of an institution in offering a guidance mechanism to bring about a resolution of the situation.

The institution definition used in this scenario is shown in Listing 5.2, which we refer to as a ‘road users’ institution. The intention is that this institution would grow to cover the management of a wider set of social interactions in road use, potentially with different institutions responsible for different contexts (e.g. a motorway road users institution and a

#### Listing 5.2: Road users institution definition

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Roadusers institution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% institution roadusers
institution roadusers;

%% component
type Agent;

%% exogenous event
exogenous event deadline;
exogenous event changeLane(Agent);
exogenous event flashLights(Agent);

%% obligations
obligation fluent obl(changeLane(Agent), deadline, vioMove(Agent));

%% institutional events
inst event iniOblChangeLane(Agent);

%% violation event
violation event vioMove(Agent);

%% rules %%
flashLights(Agent) generates iniOblChangeLane(Agent);
iniOblChangeLane(Agent) initiates perm(changeLane(Agent)), obl(changeLane(Agent),
    deadline, vioMove(Agent));

%% initiate permissions and empowerment
initially perm(deadline);
initially perm(iniOblChangeLane(Agent)), pow(iniOblChangeLane(Agent));
```

---

city road users institution). For the institution shown in Listing 5.2 the key components are where an external event `flashLights(Agent)` occurs where a vehicle flashes their lights at Agent. This generates an institution event `iniOblChangeLane(Agent)` which results in the obligation `obl(changeLane(Agent))`. When running the scenario with this institution involved, the following additional events occur:

1. At the same time as flashing its lights, the vehicle sends an institution message `flashLights(Vehicle)`.
2. The institution handles this as an external event, and creates an obligation `changeLane(Vehicle)`, where `Vehicle` is the slower vehicle causing the delay.
3. This obligation is added to Jason vehicles as a `changeLane` belief, and in this case it

triggers a lane change.

The result of this institution interaction is that the slower vehicle receives an obligation to change lane, complies, and the faster vehicle is able to get past without having to slow down.

The hypothesis of this scenario is that in the variation where there is no reaction (due to not understanding the intent) to the flashing of lights, the faster moving vehicle will eventually have to perform a hard brake to avoid colliding with the slower moving vehicle (which could result in a larger congestion shockwave pattern emerging, as considered in the scenario of Section 5.6). By comparison, with the institution active, the expectation is that there will be less (if any) disruption to the wider vehicle population, as the faster moving vehicle is able to pass when the slower vehicle changes lane.

The results of this scenario are presented in depth in Section 6.3.4, where in summary the slower vehicle receives the obligation and complies by changing lane. This results in the faster vehicle being able to move past, without performing the excessive braking shown to happen when no institution involvements occurs. In essence, it shows the institution acting to coordinate a resolution of the scenario, where by itself the slower moving vehicle had no solution (or was even aware it needed to take any action).

## 5.6 Variable speed limits

In this scenario, we explore the problem area of balancing individual pursuit versus the welfare of the wider society, as well as the issue of how to manage the autonomy of an intelligent agent. Whilst the previous scenario in Section 5.5 looked at some similar issues from the perspective of resolving an unknown behaviour through the issuance of obligations, in this scenario we consider enforcing a behaviour change on a number of vehicles in order to improve a situation for the wider collective.

The scenario context we select from the problem domain is that of a variable speed limits, where a reduction in the standard maximum speed limit for a road occurs, typically established when there is a need to slow traffic due to either accidents, lane closures, or other safety reasons. This proves an interesting area for enforcing behaviour, as the desire of human drivers to get to their destination quickly can often be seen to outweigh their conformance to the new speed restriction. Suitable detectors have to be deployed (such as variable speed cameras) to identify those not obeying the restriction, coupled with a punishment significant enough to deter drivers from breaking the rule (typically some financial penalty and addition of points to the drivers license, where enough points may cancel the license).

Considering autonomous agents in control of intelligent vehicles, there is the question of where and how should such limits be set and enforced. One approach could be to encode it in the agent, a simple `maxSpeed(V)` belief that would ensure the vehicles speed was never above  $V$ . We could hypothesis that the value of  $V$  could be updated by the road infrastructure, to account for different zones (e.g. city vs motorway) as well as implementing dynamic reductions based on a variable speed limit. However, it is also possible that such a limit may not be encoded



directly into the agent in such fashion. In the same way that humans have the capability to make decisions as to what speed to choose, it may be desirable to allow intelligent vehicles to make the same decision, for example based on weather conditions, quality of sensor data, and so on. It may be that in some cases, the penalty the agent would receive for breaking the limit is outweighed by the benefit (for example, to get a passenger to hospital). We leave such questions aside at this point, instead exploring what technological approaches can offer a solution.

For this problem, we consider the use of an institution for its ability to resolve a set of events and issue obligations for what action to take. We assume some situation has occurred, resulting in the need to establish a variable speed limit, and we wish to issue a number of vehicles with the obligation to reduce their speed. However, the institution based approach results in additional possibilities for how this variable speed limit is implemented (i.e. different speeds, targeting different vehicles), and so we need to consider what the limit is attempting to achieve, in order to select a more optimal implementation of the limit across the agents. We consider a number of studies which have looked at the impact of variable speed limits in the UK. Work by [Papageorgiou et al. \[2008\]](#) considered UK motorways, and found it difficult to make strong conclusions (other than such speed restrictions have a definite impact on flow rate) due to variables such as weather and the difficulty in repeatability of events (e.g. due to weather, peak flows on different days). In [Tafti \[2008\]](#) the M25 motorway is examined, which finds that whilst some objectives of variable speed limits have been met (smoother traffic flow, journey time reliability) others have not (no increase in peak throughput, unable to suppress shock waves).

From these works, it can be concluded that we are dealing with a complex dynamic system, and if there is only a coarse control mechanism available (e.g. all vehicles on a long stretch of road now travel below a certain speed) where there is significant variation in meeting that control signal (e.g. some vehicles travelling beneath the speed, some travelling above), with a limited closed part of the loop (e.g. the speed limit can only be a few variations, and cannot be quickly switched between these options), then it seems reasonable that outcome of using a variable speed limit does not always meet the desired state.

It is here that we see a number of benefits that intelligent vehicles coupled with a governance structure could bring, and we construct a scenario which aims to demonstrate this approach. Drawing on an example mentioned earlier in the work of [Tafti \[2008\]](#), we examine a case of undesirable vehicle behaviour known as shock waves. The triggers for such waves can be hard to identify, but we if consider the road network as a dynamic system, it can be considered that a perturbation has caused the system to enter a different (undesirable) steady state, requiring a control input to return the system to the previous (desirable e.g. uncongested) state. We represent this perturbation as a vehicle braking excessively hard for a few seconds, which on a congested road results in the vehicle behind having to brake, and the vehicle behind that having to brake, and so on. When factors such as human reaction time and over braking are included, a wave like pattern occurs, which can remain as vehicles leave the front of the wave and return to normal behaviour, whilst new vehicles arrive at the back of the wave.

One approach to dampening this wave is to enforce a variable speed limit. As this changes the vehicle speed, it implies less extreme braking would be required and that more time is available for drivers to react to events (i.e. another vehicle braking). However, (as previously mentioned) in Tafti [2008] it was observed that the variable speed limit was unable to suppress shock waves, and we put forward a number of suggestions to explain this. Firstly, that the algorithm controlling the setting of VSL may not be capable of handling this, as in Tafti [2008] this is described as being based on the measured vehicle speeds and flow levels in order to measure congestion, which is then used to activate various VSL speeds. Due to this, this is a time lag between the onset of congestion, and it being registered by the VSL algorithm (if at all) and any reduced VSL being initiated. In addition to this, there is the issue that drivers may wish to continue to travel above this new limit (in order to arrive at their destination as soon as possible) and so reduce the global conformance to, and possibly reduce the benefit of, the imposed VSL.

With this in mind, we elect to implement three variations of this scenario, all based on a vehicle performing a hard brake in order to trigger a shockwave to develop to investigate the performance of alternative approaches in managing the wave. In the first variation, no speed limit will be enforced, in order to view how the shock wave behaves when no external action is taken. In the second variation, we attempt to replicate the existing real world approach, of establishing a global variable speed limit on the affected lane.

With the third variation, we will explore assigning a speed limit to a smaller number of specific vehicles, in an attempt to pre-emptively prevent the wave from occurring. This could be extended further to attempt to clear up an existing wave, but we are interested in whether the capabilities of new technology can offer alternative approaches, and particularly whether it is possible to implement a strategy where fewer vehicles are penalised while the larger societal desire is achieved (i.e. the congestion prevented). This echoes developments which have been supported by vehicle to vehicle technology described earlier in Section 2.3.2, with work analogous to our aims in Taleb et al. [2010] considering how such technology can improve collision avoidance. In our case, we seek to use such infrastructure to send appropriate obligations to affected vehicles to reduce their speed, before their own (unmanaged) actions instigate the shockwave.

For this scenario, background traffic is inserted to build up a representative flow, and at 110 seconds a vehicle brakes hard for a duration of 6.5 seconds, before accelerating back to its previous speed and resuming normal behaviour. For the global speed limit version of this scenario, we apply a new speed limit of 22m/s (50 mph) as the braking occurs (i.e at 110 seconds). The time chosen is somewhat arbitrary, as in real life situations there is not an instant communication to the motorway infrastructure, instead there is a time lag to any VSL enforcement being set. However, the time delay in setting this VSL could itself become part of the investigation, drawing away from the main investigation into issuing such speed guidance via an institution. Whilst this delay time value is somewhat arbitrary in this scenario, it does introduce the temporal element around this obligation, both at what time it should come into effect, and how long it should last for. Furthermore, there is the question of whether having

diverted the vehicle from its steady state for a period of time, whether some corrective action should be implemented to return the vehicle to a ‘normal’ state (with consideration for how long such corrective action should take). Whilst these issues are not addressed in this scenario, they are considered further in the future work discussions of Section 7.3.

For this scenario, the institution used is shown in Listing 5.3, and is responsible for generating speed restrictions to specific individual vehicles in the following process:

1. A Vehicle `jasonCar1` performs a hard brake at 110 seconds into simulation run.
2. At the same time that the braking begins, `jasonCar1` two messages to the institution.
3. The first message sets some initial conditions of the form `vehiclePosition(VehicleName, PositionNumber)` in the institution, for the vehicle names behind `jasonCar1` and their relative position number, `p1`, `p2`, `p3`, for the three vehicles behind `jasonCar1`.
4. The second message is an event to the institution of the form `emergencyBrake(jasonCar1)`, to inform the institution that an emergency brake has occurred.
5. The institution issues obligations of the form `slowDown(Follower, Speed)` to each of the vehicles it was informed about as being behind the braking car. The `Speed` component varies from `slow` for the vehicle immediately behind, to `mediumSlow` for the next vehicle, and `medium` for the rest.
6. On receipt of this obligation, the Jason framework translates this to a `slowDown(Value)` belief update for each affected agent.
7. Agents identify the appropriate speed for this based on their beliefs: `speedModifier(slow, 5)`, `speedModifier(mediumSlow, 12)`, and `speedModifier(medium, 20)`.
8. Following this, agents set their speed to the determined value.

This institution introduces an additional aspect of action resolution to be performed by targeted vehicles, where values for speed are left in non specific terms (e.g. `slow`, `medium`). Vehicles now have the responsibility to decide whether to implement the obligation, but also to transform the abstract term (i.e. `slow`) into a real term (i.e. 5 meters per second). This highlights the nature of the the relationship between institutions and vehicles in this scenario, rather than being a prescriptive enforcement, it serves as a guidance mechanism where vehicles implement obligations according to their current situation (which they have better awareness of than the institution).

For this scenario, the hypothesis is that with appropriate vehicle parameters and scenario configuration, a shockwave will form in the no VSL variation when `jasonCar1` performs the hard brake. With no management of this shockwave, it should remain persistent during the duration of the simulation, and be observable in captured metrics. With the global VSL approach, the expectation is that a decrease across all vehicle speeds should be observable, but the impact of this on any congestion behaviour is difficult to predict. It is possible that by

Listing 5.3: Variable Speed Limit institution definition

```
%%% institution vsl
institution vsl;

%%% component
type Agent;
type Follower;
type Position;
type Speed;

%%% exogenous event
exogenous event deadline;
exogenous event emergencyBrake(Agent);
exogenous event slowDown(Follower, Speed);

%%% institutional events
inst event iniOblSlowDown(Agent);

%%% obligations
obligation fluent obl(slowDown(Follower, Speed), deadline, vioSlowDown(Follower));

%%% violation event
violation event vioSlowDown(Follower);

%%% fluents
fluent speedMode(Follower, Speed);
fluent vehiclePosition(Follower, Position);

emergencyBrake(Agent) generates iniOblSlowDown(Agent);
%% if vehicle is p1 i.e. directly behind, slow it down alot
iniOblSlowDown(Agent) initiates perm(slowDown(Follower,Speed)), obl(slowDown(
    Follower,slow), deadline, vioSlowDown(Follower)) if vehiclePosition(Follower,
    p1);
%% if its in p2 slow it down to mediumSlow
iniOblSlowDown(Agent) initiates perm(slowDown(Follower,Speed)), obl(slowDown(
    Follower,mediumSlow), deadline, vioSlowDown(Follower)) if vehiclePosition(
    Follower,p2);
%% if its in p3 slow it down to medium
iniOblSlowDown(Agent) initiates perm(slowDown(Follower,Speed)), obl(slowDown(
    Follower,medium), deadline, vioSlowDown(Follower)) if vehiclePosition(Follower
    ,p3);

%%% initiate permissions and empowerment
initially perm(deadline);
initially perm(iniOblSlowDown(Agent)), pow(iniOblSlowDown(Agent));
initially speedMode(centralMember2, normal);
initially speedMode(centralMember3, normal);
initially speedMode(centralMember4, normal);
```

forcing vehicles to travel at a lower speed, that any excessive braking and acceleration may be mitigated by an increased time to react, but this will be assessed based on the experimental results. The hypothesis regarding the institution approach is that only a small subset of the vehicle population will show a decrease in speed, and that some impact on the congestion behaviour may be observable. However, it is difficult to know whether this impact will be positive or negative, as there are a number of parameters involved (e.g. number of vehicles to target with VSL, how significant the speed reduction should be, how long it should last for) which may be the subject of further tuning.

The results for this scenario are presented in detail in Section 6.3.5, where in summary whilst the institution is shown to have the expected effect on vehicle speeds, the global impact becomes difficult to interpret and may require further work to tune parameters of the institution obligations (i.e. number of vehicles targeted, how long the obligation should persist for, and how significant the speed reduction should be).

## 5.7 Post accident management

In this scenario, we investigate the management and coordination of multiple events, introducing elements where vehicle agents are required to interact with each other to resolve a complex situation, raising similar management of autonomy questions as posed in the scenario of Section 5.6, as well as drawing in additional bodies to widen the scenario outside of purely vehicle to vehicle interactions.

We draw inspiration for the scenario context here from details in a recent press release (Vilkas [2014]) that from October 2015, all new vehicles and vans will be fitted with an electronic device which automatically informs the emergency services when a vehicle crashes. This system is referred to as “eCall”, and makes use of cellular communication as the means to relay GPS coordinates of the crash back to an emergency service provider. In Pinart et al. [2009] implementation options are presented, with further extensions to the basic message contents expanded upon to include information such as the number and medical records of passengers, severity of the crash, and so on.

From this, we have one aspect of the scenario established, that of relaying information about the crash to the emergency services. Combined with this, we explore the capability shown in the scenarios outlined in Sections 5.6 and 5.5, of improving the coordination of intelligent vehicles, in order to resolve the post-accident coordination required where vehicles obstructed by the crash can change lane to move past the accident.

In Figure 5-6 the general concept of this scenario is illustrated, where the far left vehicle (shown in green) has crashed, and is now stationary in its lane. Two Jason controlled vehicles are immediately behind this crashed vehicle, and have now become stuck, as they have been unable to change lane due to the presence of other traffic. It is in the management of this problem that we draw on the use of an institution, to manage pairs of coordinating vehicles where one vehicle needs to slow down to ensure sufficient space in front of it, into which the other vehicle can pull out and pass the obstruction of the crashed vehicle.

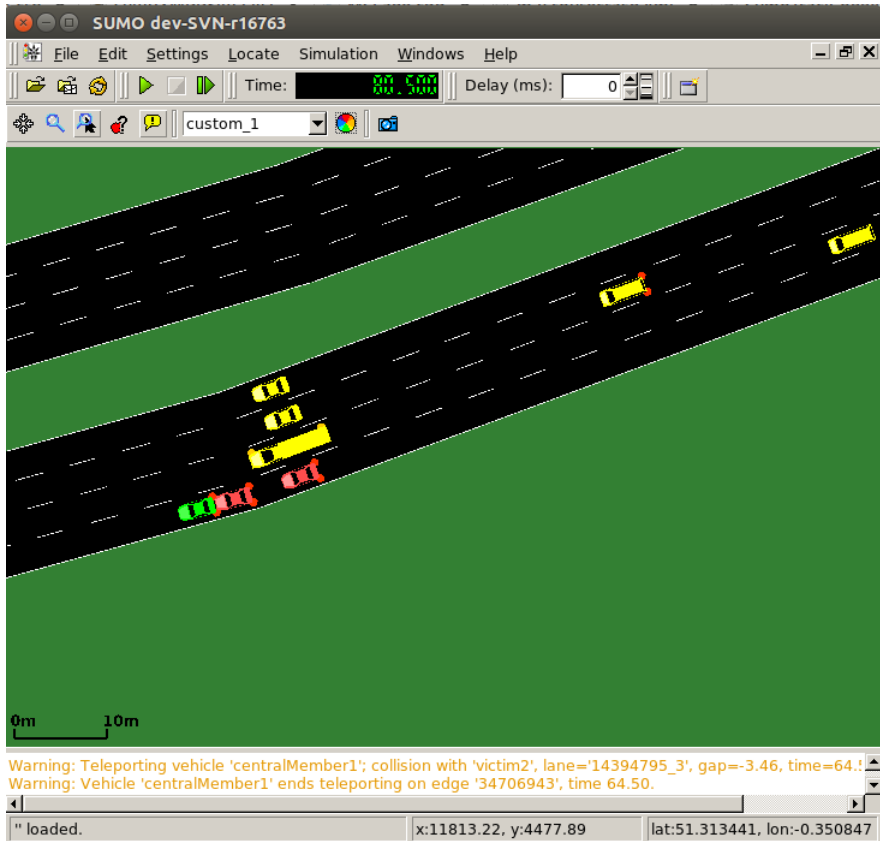


Figure 5-6: Vehicle motorway crash scenario

We also extend the use of institutions further in this scenario, and draw on multiple institution interactions to manage the wider set of issues caused by such a crash, and illustrate this through the use of an ‘insurance’ institution. Here we illustrate both domain relevance (i.e. the eCall system may communicate directly with insurance companies) and also the broader relevance, with an external body dealing with financial penalties which could be contrary to the desire of an individual agent.

For this set of interactions, four separate institutions have been constructed, which are able to share events through the bridge architecture discussed earlier in Section 4.1.4. We now introduce these institutions, with links to their institution definitions found in Appendix A:

- Crash group: This institution is established to reflect the group of vehicles involved in the crash, mainly focussed on vehicles who may need to merge to another lane. The institution definition is provided in Listing A.1.
- Emergency services: This institution is used to coordinate the response of emergency services. The institution definition is provided in Listing A.2.
- Insurer: This institution is responsible for managing insurance claims between affected parties. The institution definition is provided in Listing A.3.
- Motorway management: This institution has the authority to enforce any required co-ordination, such as the merging of vehicles. The institution definition is provided in Listing A.4.

Following on from the definition of the institutions used, we now outline the baseline version of this scenario, with no institution involvement, where the events are:

1. A Jason vehicle `jasonCar1` approaches a slower vehicle, and for some reason collides into the rear of the slower vehicle.
2. `jasonCar1` comes to a halt in its motorway lane, whilst the other vehicle drives away.
3. There are two further Jason vehicles behind `jasonCar1`, and three in the adjacent lane.
4. The Jason vehicles use some native SUMO behaviours to provide lane changing functionality, and so attempt to change lane to maintain speed.
5. These vehicles will be delayed behind the crashed `jasonCar1`, but eventually be able to change lane and continue on their journey.

For the variant of this scenario with multiple institutions active, a more complex set of events occurred:

1. On being informed it has crashed, the Jason agent vehicle `jasonCar1` informs the crash group institution of a set of initial fluents, of the form `mergePairing(VehicleToMerge, VehicleToPermitMerge, CrashedVehicleName)`.
2. The crash group initiates a compensation claim from the victim against `jasonCar1`, and initiates requests to merge from the initial `mergePairing` details.

3. The compensation claim cross generates an event to check the insurance of `jasonCar1`, cross initiates `payee(Victim, jasonCar1)`, and cross generates a series of `performMerge` based on the initial fluents of merge pairings.
4. The insurance institution issues an obligation to pay the `Victim`, based on the `payee` information.
5. The insurance institution contains the fluent that `jasonCar1` has no valid insurance, which causes the insurer institution to cross generate a `crimeCommitted` event in the `emergencyServices` institution.
6. Three obligations are generated as a result of the above process, two merge obligations from the `motorwaymanagement` institution, and a `dispatchUnit` from the `emergencyServices` institution to attend the scene.

The hypothesis of this scenario is that in the domain context we should see non-optimal merge behaviour when there is no institution active, resulting in vehicles becoming obstructed and delayed by the crash vehicle when they are unable to change lane. In comparison, with the institution active the aspiration is that by coordinating merge behaviour between intelligent vehicles, that there should be less delay in navigating past the crashed vehicle, as the institution removes the social ambiguity of being able to change lanes, and instead manages this through appropriate obligations.

The results of this scenario are presented in Section 6.3.6, where in summary an improvement in the domain performance is shown, due to the institution activity in coordinating a lane merge behaviour, which allows vehicles to navigate around the obstacle of the crashed vehicle, and demonstrates the functionality where multiple institutions may interact in order to address the problem statements set out in Section 3.2.

## 5.8 Summary

In this chapter we have detailed the scenarios which aim to explore the problem statements put forward in Chapter 3. Firstly, the problem area of communication and understanding is investigated through scenarios which explore elements of Situational Awareness, using the context of a vehicle convoy moving to a destination, exchanging first perception level information and then a higher level of knowledge exchange of waypoint information. The topic of situational awareness is explored further, focussing on the projection element of SA, with a scenario based around traffic light states and the impact that their future state would have on a vehicle. This scenario also introduces the use of institutions as a governance structure, where vehicles receive an obligation to reduce their speed in order to arrive at a traffic light whilst it is green.

We then move onto a number of more complex domain scenarios, and whilst we remain mindful of exchanging information at appropriate levels (in SA terms), we focus on the problem statements of Section 3.2 concerned with resolving problematic situations and balancing individual pursuits versus greater social welfare. A scenario based on a vehicle



wishing to overtake another, where this desire is expressed through a flash of headlights is put forward, with the use of an institution providing an external resolution of what the culturally and contextually dependent flash of headlight means. This is followed by a scenario which investigates the use of an institution in the enforcement of variable speed limits, which also explores whether targeting obligations to a smaller number of vehicles can achieve the wider goal (in this case, reducing congestion), and compares this against using no speed limit as well as a global speed limit where all vehicles are affected.

The last scenario put forward is based around post-accident management, where the use of external institutions is explored further as a means to handle the response to such an event, both in the immediate coordination (of allowing vehicles to merge lanes in order to get past the obstruction) and the wider responses and actions required (e.g. dispatching an emergency vehicle and processing insurance claims). We also consider this as a mechanism to improve the overall shared situational awareness of the collective, with institutions performing the role of mediating intent (i.e. in supporting the merge lane behaviour where one vehicle wishes to pull out in front of another). Whilst this scenario could be expanded significantly, we focus on demonstrating the capability and developing a generalised framework appropriate for this genre of problems, in line with the motivation outlined in Chapter 3.

In this chapter, we have presented the scenarios which we use to explore the problem statements of Section 3.2. Communication is explored using various levels of SA, in an effort to improve both the individual and collective understanding of the environment. Institutional frameworks are introduced to offer assistance to intelligent agents in the form of guidance (when the agents are unsure what action to take) and as an enforcement mechanism (when agents pursue goals contrary to the benefit of the wider society). These scenarios build on the capability put forward in Chapter 4 and demonstrate the application of this generic framework to specific vehicle scenarios relevant to real world problems, whilst highlighting forms of capability which are applicable to other domains.

## Chapter 6

# Analysis of results

A discussion of general problems faced by artificial socio-cognitive systems is presented in Chapter 3, where a number of problem statements are put forward to be explored in the specific problem domain of intelligent vehicles. A number of scenarios have been constructed (as detailed in Chapter 5), which are implemented using the architecture discussed in Chapter 4. These scenarios are used to demonstrate findings based in the intelligent vehicle domain, but with application to the wider generalised problem areas.

In Section 6.1 we present a number of general findings arising from the use of the framework, where we find the use of inconsistent units across distributed components causing errors, insufficient management of the intelligent agent belief base causing performance issues, and message exchange taking place at too high a rate resulting in plan failure in agents. Following this, in Section 6.2 we present details of communication and domain specific metrics which have been used in the course of investigating each scenario.

We then present and discuss the findings from the scenarios of Chapter 5. In Section 6.3.1 we present the results of varying the quantity of low level SA message exchange when performing a vehicle convoy activity, finding no impact on domain performance whilst seeing a substantial decrease in message volume. In Section 6.3.2 we show the impact of replacing the low level messages with a higher level message set containing details of the convoy route, and find whilst communication quantity is similar, domain performance improves, showing less acceleration and braking. In Section 6.3.3 we present results of SA being explored further, where high level projection of the future environment state is used and vehicles modify their speed in order arrive at traffic lights whilst they are green, resulting in an improvement in fuel consumption. In Section 6.3.4 we present results of an institution being used to assist an intelligent agent where the agent is unaware it needs to take any action when another vehicle flashes its headlights, and show the success of the institution in resolving this situation. In Section 6.3.6 we present an alternative institution use, in this case enforcing speed reductions to a small number of vehicles in order to bring about the wider social desire of reducing congestion. Whilst this scenario demonstrates the institution, we suggest further tuning of the precise implementation of the obligations (e.g. duration and value of reduced speed) is required. Finally, in Section 6.3.6 we

present results where multiple institutions are shown interacting in order to manage a post-accident scenario, and we show the domain benefit in the merge-lane coordination this provides.

A number of these findings have relevance to the broader high level set of problems, and in Section 6.4 we present a generalised analysis of the overall findings of the work, where we reflect on the general problem areas put forward in Section 3.1. A review of the findings in relation to the problem statements and objectives follows later in the conclusions Chapter 8.

## 6.1 Preliminary work

When constructing the first scenario, a number of issues were encountered which are not an obvious fit for either the previous discussion in Section 5.1.3 around low level system testing, or the discussions which follow later in the chapter regarding scenario-specific findings. They could be considered as the result of ‘shake down’ tests, with the framework being used in various configurations during the development of scenarios and where unexpected or initially inexplicable events have occurred, but also as cases where generalised issues have been found which inform the configuration of the framework, as well as aspects of the problem statements of Section 3.2.

We present a number of these findings here, where they relate to general issues of the problem space rather than specific scenario outcomes, as unexpected but beneficial results which add to our understanding of the area, highlighting both generalised problems but also the capability of the framework to support investigation into underlying issues.

### 6.1.1 Incorrect geospatial calculations

This collection of issues relates to the use and manipulation of geospatial data such as vehicle positions, their orientation, and speed. Since messages are exchanged in RDF format, this alleviates some issues, as vehicle position data is communicated in a standard format and consistent unit measurement. However, as various system components use their own coordinate systems, significant numbers of conversions are required. Cases encountered have included incorrect assumptions about units used (e.g. radians versus degrees) and the accuracy of libraries, leading to unexpected (e.g. geospatially incorrect or inaccurate) output.

An example of this type of problem is in the calculation of collision volumes. The Jason driver agent is responsible for calculating this volume, using current orientation and speed to predict occupied space for the next two seconds. If any objects are detected within this volume, then the vehicle performs a stop. The position of the volume was being calculated correctly, but its orientation was incorrect, resulting in unexpected stops occurring. The observed behaviour was that during a convoy move, vehicles would occasionally stop unexpectedly. Such a behaviour can occur for a number of reasons, for example, no new position is received from the convoy lead vehicle, therefore agent believes it has arrived at destination and comes to a stop.

Figure 6-1 illustrates various uses of collision volumes. There are three numeric annotations, number 1 indicates data derived from the Jason driver agent as to the current desired location,

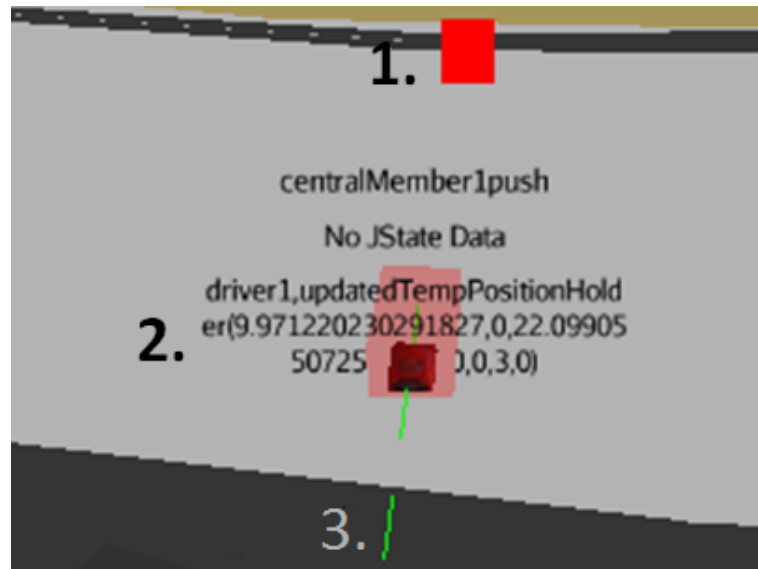


Figure 6-1: 3D view augmented with collision volume data

indicated by a red box. Number 2 shows a red volume projected over the vehicle, which is the current collision volume as calculated by the agent. Finally, number 3 displays green lines that are used to show previous collision volume start and end points. The effect of these three additions has been to facilitate rapid debugging of not just collision data itself, but also an aid in the explanation of emergent convoy behaviour in other scenarios.

### 6.1.2 Incorrect management of agent belief base

This issue relates to the agent layer, specifically the handling of belief updates received from the environment via the middleware. The simulated vehicles broadcast their position data, which is then received via the extended Jason environment class, and added to each agent's belief base as a new belief e.g. `info(position,heading)`. As soon as position or heading values change, a new `info()` belief is added. However, the previous `info()` belief should be removed (as the vehicle is no longer at that location). Any agent plans that react to `info()` beliefs are only informed about the latest update, so these unwanted 'historical' `info()` beliefs do not cause any apparent problem. However, while investigating performance issues, initially the simulation behaved as expected, but the likelihood of unexpected behaviour (most commonly, route planning ceased) appeared to increase with longer simulation runs.

Profiling showed resources consumption did grow over time, and the rising number of beliefs in the agent was a contributing factor. This experience underlines the need to be able to observe all parts of the system, such as in Figure 6-1, from the 3D view, which shows summary information extracted from the Jason mind state. The text to the right of number two displays the most recent message received by the driver agent, and can also display the current belief count. Consequently, it is relatively easy to correlate increasing belief count with declining

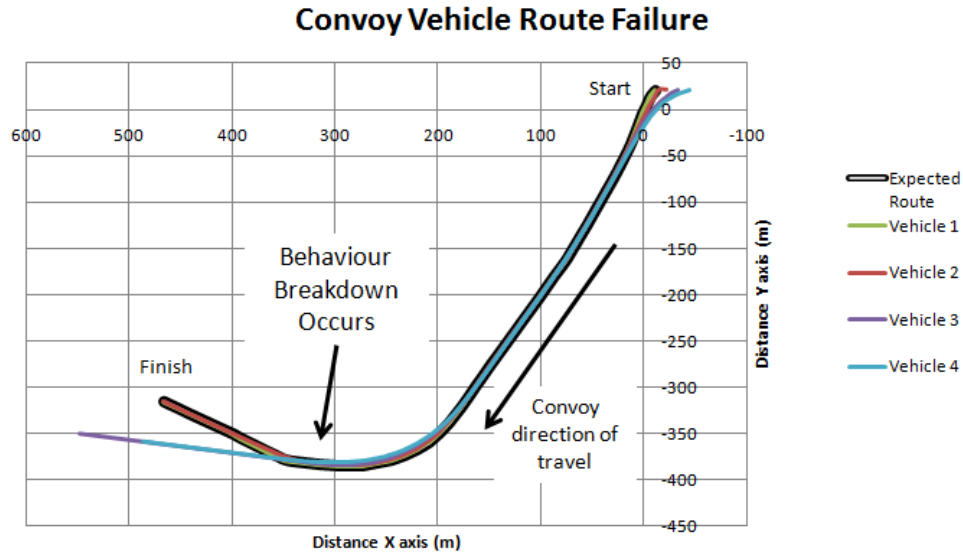


Figure 6-2: Case of convoy failure during data saturation

performance and hence direct debug effort appropriately. Such component health indicators are generically useful, since out of the ordinary – whatever that is defined to be – belief counts can be symptomatic of fundamental problems.

### 6.1.3 Inappropriate data transmission rates

The subject of appropriate transfer rates was introduced in the problem statements put forward earlier in Section 3.2. In practice, such ‘impedance’ mismatches have proven to be one of the most challenging to debug, as each individual component has been subjected to functional testing, but this is a ‘non-functional’ property. For a reactive system to be capable of responding quickly to unfolding events, there is a motivation to increase update rates to make data as timely as possible. However, this ignores what is useful and appropriate for a given component. For the intelligent agent layer, ideally the quantity should be low but with high information value.

The impact of simply increasing the data rates is shown in Figure 6-2 with an example of failure in vehicle convoy movement due to data saturation. Here, vehicles 1 and 2 arrive correctly at the destination, vehicle 3 has a failure which results in it no longer following the convoy direction. Vehicle 4 continues to follow 3; correct according to convoy-following behaviour but incorrect in terms of the overall convoy goal. Investigation revealed that vehicle 3 is suffering from data deluge: there are too many position updates for it to process before the next time step occurs, plan selection for the next position to move to fails, and there is no recovery mechanism from this state (so the vehicle continues in a straight line). In the longer term safety behaviours to deal with such situations can be developed, but the underlying issue of managing data flow effectively needs tackling first. It is also appropriate to reiterate the potential utility of stream reasoning (as discussed earlier in Section 2.1.2) to contribute to

solving this problem.

In summary, we have found three cases where small mistakes, incorrect assumptions or inappropriate design choices have had a negative impact. In the first example, an incorrect assumption of units lead to incorrect spatial calculations, and we see this as highlighting a benefit in adopting semantic annotation in message exchange, where the additional definition of the message content would assist the recipient in determining the units used. In the second example, we see an agents belief base growing increasingly large which leads to performance issues, which relates both to the need to communicate at appropriate levels of SA to reduce information overload, and a need to consider knowledge management within the intelligence layer as to how much information is needed and should be retained. Finally, in the third example we see a further example of the need to communicate at appropriate levels, where too much information has caused a failure in plan execution within an intelligent agent.

## 6.2 Metrics

The capability for metrics capture was introduced earlier in Section 4.1.6, as part of the monitoring capability provided by the BSF set of tools, comprising of both infrastructure metrics (focussing on communication) and also various vehicle metrics (focussing on the problem domain). We present the various groups of metrics used here as they themselves form part of the analysis; as there is no standard measurement to use, it depends both on the domain and the scenario itself. Some have generic applicability, such as communication metrics, whereas others are more bespoke. Since we can re-run scenarios, an evolutionary approach was adopted with a number of these metrics, developing additional features or measurement approaches where promising data was found but not reflected clearly in the existing metric(s). Here we present the metrics developed over the course of the investigative scenarios, and their interpretation.

### 6.2.1 Communication metrics

With an interest in information exchange from the outset, there was a need to develop suitable metrics to display how many messages are being published via the framework. As described earlier in Section 4.2 there were initial problems with the capacity of the system to exchange a high number of messages reliably, but even with this bottleneck improved, there is still a need to consider this metric as core to the reliable performance of scenarios. The work in Section 4.2 went some way in defining the ‘performance envelope’ of the system used in this experimentation, and if scenarios creep outside of this known area then there is concern as to how various components may behave.

Primarily this metric is used in the perception exchange scenarios, where there is interest in reducing the amount of communication whilst still achieving domain performance in vehicle convoy behaviour.

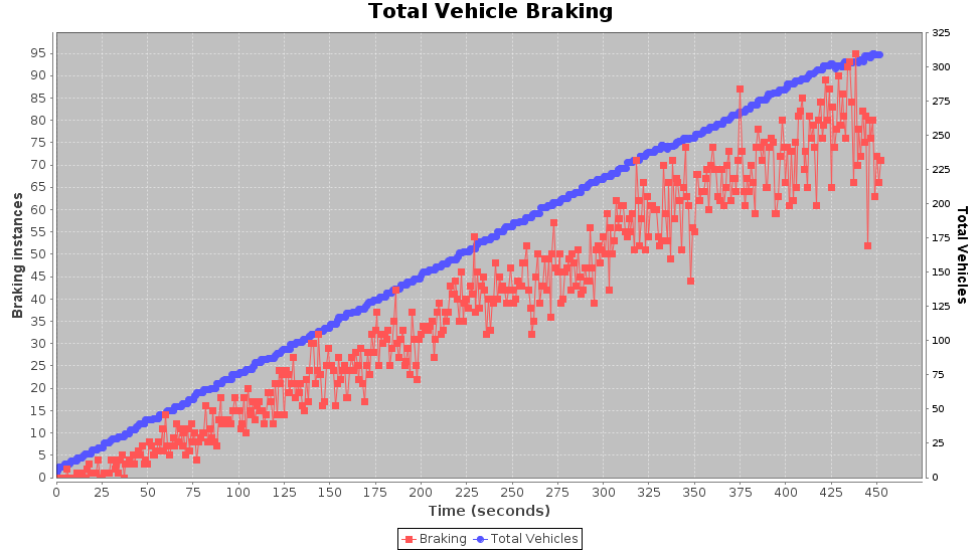


Figure 6-3: Measurement of vehicle braking

### 6.2.2 Vehicle performance metrics

When considering the impact of differing scenarios on vehicle behaviour, a set of metrics is required to capture performance indicators at the individual vehicle level. Some of these measures may be specific to the requirements of scenarios (e.g. a traffic calming scenario may wish to measure vehicle speeds) whereas other may be generic and of interest to a wider range of scenarios (e.g. fuel consumption). Furthermore, some metrics (such as fuel consumption) require specific modelling in order to be represented, whereas other just require the information to be published from the simulator (such as vehicle speeds).

For the scenarios in this work, there are three main metrics which have been used. The first is vehicle speed, which can be simply obtained from the SUMO simulator and included in vehicle RDF messages in order to allow subscriber to either store in a database for off-line analysis, or displayed in realtime via the BSF tools. The second is fuel consumption, with emissions included in this category, as they are closely linked. The SUMO simulator draws on the Handbook Emission Factors for Road Transport (HBEFA) <sup>1</sup> to implement a model representing emissions of pollutants (e.g. CO<sub>2</sub>, CO, etc) as well as fuel consumption. Finally, the third metric developed reports braking and acceleration performed by each vehicle. The objective with this metric is to indicate the occurrence of congestion, as there may be an increase in braking and acceleration during congestion, rather than when situated in a more steady normal flow of traffic.

However, when the metric of braking as an indicator of congestion occurring was explored further, it was found to be more closely correlated to the number of vehicles in the simulation than any specific events occurring. This can be seen in Figure 6-3, where the number of vehicles

<sup>1</sup><http://www.hbefa.net/e/index.html>

in the simulation is plotted along with the number of occurrences of braking, at each time step. For this reason, an alternative measurement of congestion was sought, and this is presented later in the motorway metrics of Section 6.2.4.

### 6.2.3 Convoy performance metrics

Three scenarios are based around the topic of vehicle convoys, where one vehicle performs the role of ‘convoy leader’ and a number of other vehicles are expected to following behind in some formation. For these scenarios, some specific metrics were developed in order to measure how the convoy members have performed in following the lead vehicle.

The first metric focusses on the deviation from the lead vehicle’s route. If the x,y coordinates of the lead vehicle were plotted as a series of joined lines, then for each following vehicle, for each x,y coordinate the deviation from the leader’s plot can be calculated. This result can be used either as an instantaneous value, or cumulative to compare the overall performance of different convoy strategies.

The second metric attempts to measure the ‘cohesion’ of the convoy, considering the distance between the convoy leader and following vehicles. The previous deviation metric could be considered as measure a lateral deviation, with this metric of ‘distance to the leader’ providing a longitudinal measurement of convoy performance.

Whilst this metric gives an indication of the convoy performance, it is based on the assumption that this performance can be reflected in terms of distance from the convoy leader and the convoy leader’s route. Whilst this is likely to hold true for the scenarios we explore, there may be situations where this is not the case, such as if the convoy has to split or join with another convoy. There may also be additional components which improve the measurement of convoy performance, such as how much convoy members are braking and accelerating. However, for the scenarios we explore, the existing metric provides an indication of performance, and considering the problem statements of Section 3.2 we wish to avoid over developing these metrics and in doing so, become overly domain specific.

### 6.2.4 Motorway metrics

An additional metric was developed primarily for the motorway based scenarios, where there was a need to represent a more ‘global’ view of the scenario than the vehicle-centric metrics. The main requirement for this metric is to identify where there may be some deviation occurring from the ‘normal’ state of the motorway vehicles. In order to identify the ‘normal’ conditions, the metric representation needs to show the state of all vehicles, such that vehicles deviating significantly from the norm can be identified.

The approach adopted is to capture all vehicle speeds, along with the gap ahead of each vehicle (to the vehicle in front of it in the current lane). This provides the x-axis component, whilst the vehicle’s position along the planned route is used to form the y-axis component. Therefore, if there are a number of vehicles affected by a slow moving vehicle, there may be a cluster of reduced speed values plotted at similar route position points. However, the drawback



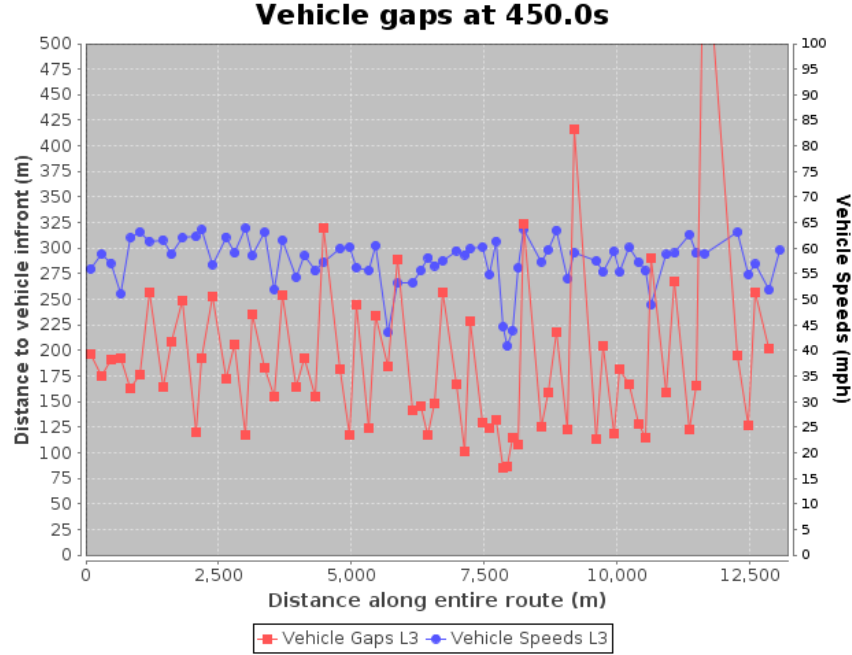


Figure 6-4: Measurement of convoy gaps and speeds

to this approach is that for each simulation time step, the graph needs to be replotted and so the only way to view the time dimension is to view the plots generated either in real-time or as a video post scenario run.

An example of such a plot is shown in Figure 6-4, with the simulation state shown after 450 seconds. For this plot, we have extracted only data from Lane 3 (referred to in the graph as L3) in order to avoid the presentation becoming cluttered with results from other lanes. From the left-hand side of the plot, we have plot points at 0m along the route (i.e. the last vehicle), for a vehicle travelling at approximately 55mph, where the gap to the vehicle ahead of it is 200m. Near the 'front' of the motorway traffic approximately 13'000m along the route there is a set of plot points for a vehicle travelling at approximately 50mph with a gap to the vehicle ahead of 200m. Slightly further along the route a single plot point for a vehicle can be seen travelling at 60mph, but with no corresponding plot point for gap distance as there are no vehicles ahead of this.

Figure 6-4 is used as an example, with no specific intentional events occurring, though on examination there appears to be one area where some deviation from typical behaviour has taken place. At approximately 8000m, there are a cluster of plot points showing both reduced speed (40mph to 45mph when the average is around 60mph) and also reduced gap ahead values of around 100m. This provides an example of the capability of this metric, and further investigation of what is occurring with this set of vehicles could now follow if this were being used in the research scenarios.

Whilst this metric provides a view of the overall vehicle population, as it represents the

speed and gaps at a particular time sample it proves difficult to display emerging behaviour, and transient events may mislead interpretation. For example, in Figure 6-4 the cluster of slower moving vehicles could be due to a vehicle changing lane and causing vehicles to brake briefly. If this behaviour only lasts a few seconds, then it may not be of concern, but if it remains for a long period then it may suggest some congestion pattern has emerged. We have developed the capability to construct videos from this graph (as used in the scenario videos presented in Section 6.3) which assists with analysis but not for presentation as a single graph. There is potential that an enhanced metric, based on a derivative of the existing approach, may offer a solution. This isn't pursued as we are able to show the impact between scenario approaches with the existing metric, and we wish to avoid the distraction of chasing higher fidelity metrics for this specific domain when our interest lies in demonstrating a generalised approach.

## 6.3 Analysis of scenarios

In this section we present analysis of each scenario-specific set of results. The goals of each scenario are re-introduced, along with discussion of required metrics (e.g. as outlined earlier in Sections 6.2.1 and 6.2) to assess the performance against the goals of each scenario. Following this, discussion and interpretation of the results is put forward for each specific scenario, with a general discussion following in Section 6.4.

### 6.3.1 Perception driven

In this scenario, as discussed in Section 5.2, we are interested in assessing the performance of a vehicle convoy, varying communication strategies between a data push of perceptions versus a data pull approach. Drawing on the metrics put forward earlier in Section 6.2, we present findings on the variation in communication impact, as well as considering the performance of the vehicles using domain metrics.

In the scenario presented here, vehicle simulation uses the xmppVehicle representation (as discussed earlier in Section 4.1.3), rather than the SUMO traffic simulator. Whilst it would be possible to (re-)implement these scenarios in SUMO, the approach of moving to specific x,y locations rather than specifying routes is contrary to the SUMO design for route-based navigation. The scenario follows the approach set out in Section 5.2, where a convoy leader is moving to a specific destination, and convoy followers move to the last known position of the vehicle ahead of them. We explore data push (where the vehicle ahead broadcasts its position to the vehicle behind) and a data pull (where the vehicle behind asks the vehicle ahead its position, at a lower frequency than the push) approaches, and assess the performance differences between these. Firstly we present the communication impact to inform our view on the problem statement of Section 3.2 regarding quantity and appropriateness of information exchange, and this is followed by reviewing the impact on domain performance.

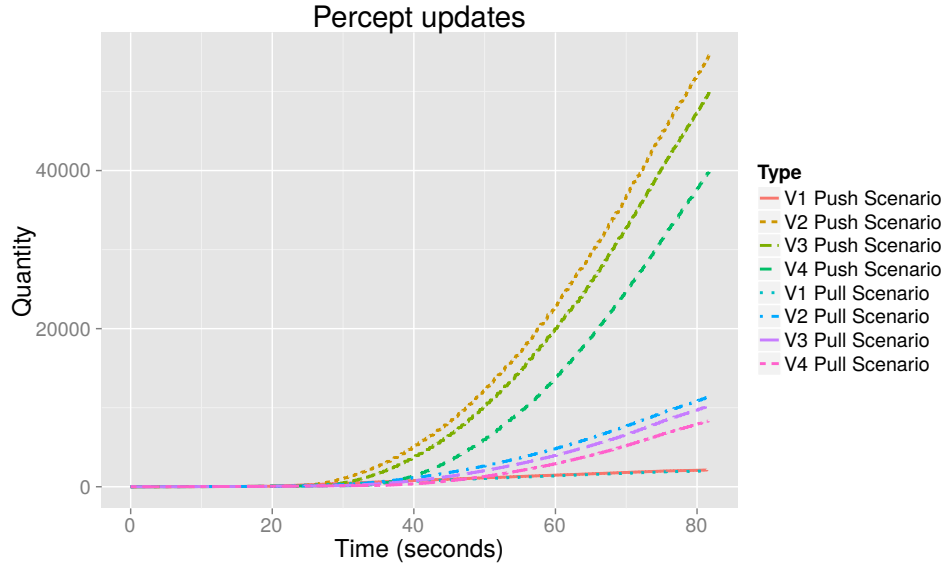


Figure 6-5: Percept count over time

## Communication

We expect there to be a reduction in the amount of communication when the data pull approach is adopted, but we need to quantify this impact before assessing any positive or negative affects on the convoy behaviour.

In Figure 6-5 the captured percept update quantities are shown, where we can see three distinct groups:

1. Vehicles V2, V3 and V4 in the push scenario. These show the highest quantity of percept updates, with V4 having approximately 40'000 updates, and V2 and V4 approximately 50'000.
2. Vehicles V2, V4 and V4 in the pull scenario. Here we see a much lower quantity of percept updates, all close to approximately 10'000 updates.
3. V1 in both push and pull scenario. In both scenario variations, V1 shows the same quantity of approximately 2'000 percept updates.

Considering these results, the convoy leaders show similar, low, update quantities in both scenarios, which is reassuring as these vehicles are receiving no updates from other vehicles, instead this quantity increases linearly over time most likely due to basic percept updates such as (self) vehicle position information. This also forms a baseline as to how the other three vehicles in the convoy would behave without the additional overhead of their convoy communication.

This then leaves the two distinct groups of data push versus data pull, with the data push having in the order of five times the communication volume. We don't seek to over labour this

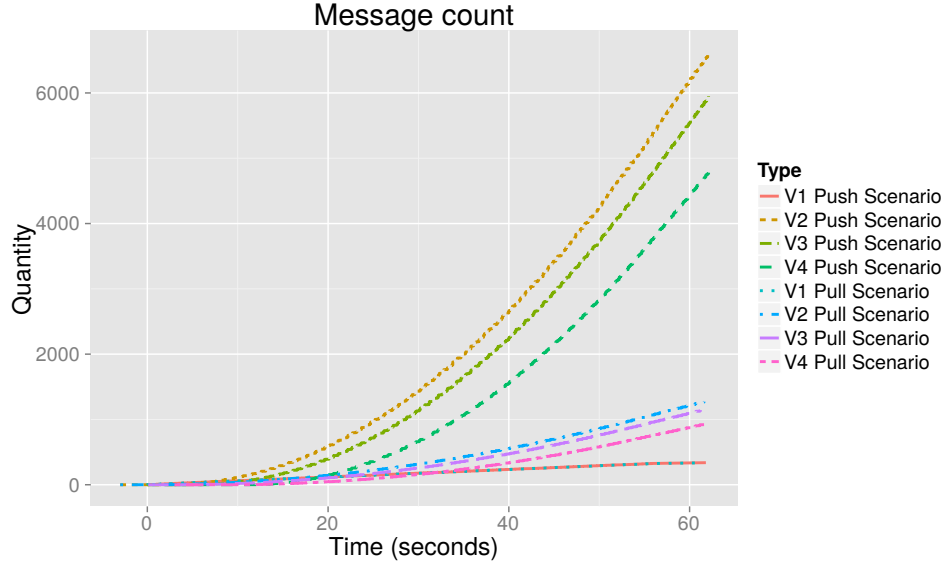


Figure 6-6: Message count over time

finding, as it follows that the data pull would be expected to have a lower volume since it is communicating less frequently. An additional point to consider is that the approach of how frequently to request information via the pull approach is quite crude, and with tuning it may be possible to reduce the quantity further and still maintain a cohesive convoy. The general topic of tuning is revisited in the further work of Section 7.1.4, as for this scenario we have the capability to perform multiple simulation runs whilst varying the communication parameters, an appropriate metric to measure the effect of the parameter changes, and a BSF module could be developed to identify alternative (possibly improved) parameter values.

In Figure 6-6 the quantity of message exchange between agents is shown. Similar groupings to those identified in Figure 6-5 can be seen: the two lead convoy vehicles show low linear levels of message exchange, a grouping of three vehicles from the pull scenario show an increased level of message exchange, and finally a grouping of three vehicles from the push scenario show much higher quantities of message exchange.

Similarly to the discussion of perception quantities, this is due to the increased frequency of communication in the push strategy, and in fact as the profile of the results in both Figure 6-5 and Figure 6-6 is similar, it would seem that there is a correlation between percept updates and the quantity of messages exchanged between agents. This seems justified, as one message may be causing multiple percept updates, but it highlights the need to consider efficiency when constructing intelligent agents. Earlier in Section 6.1 the effect on insufficient plans and mismanagement of the belief base was discussed, and now we have a potential case where there may be inefficiencies in plan design causing greater perception updates than necessary. In general terms, there are often opportunities to improve code efficiency but there is a balance

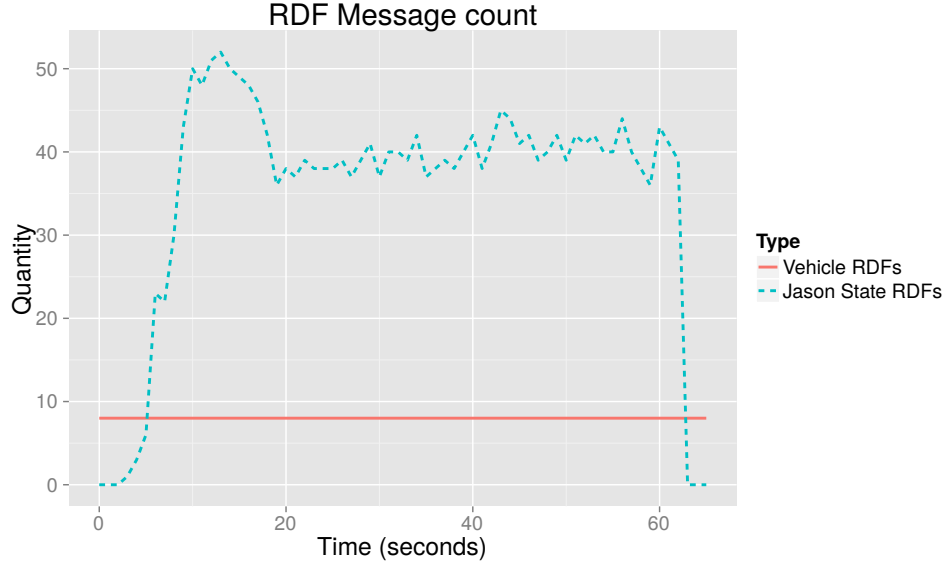


Figure 6-7: RDF Message count over time

to be found between the necessity (i.e. is there a memory leak) versus the desirable (i.e. could less memory be used).

In Figure 6-7 the message quantity viewed from the BSF framework perspective is shown, using the `rdfMonitor` tool where the number of received RDF messages per second is captured. It is worth noting the difference shown in message quantity here compared to the earlier Figure 6-6, i.e. the comparison between externally published messages on the BSF versus internal Jason agent message exchange. Whilst some Jason state data is extracted and published to the framework (shown as Jason State RDFs in Figure 6-7) this is only a subset, and it can be seen that much higher communication is taking place internally within the agent platform. This draws attention to two aspects: that each BSF component can process large volumes of data internally (if required) and manage their communication to the framework in some unsynchronised fashion. It also raises the question of observability, that whilst certain internal (i.e. Jason agent state data) information is published to the BSF and is therefore observable, how much data is it appropriate to publish?

In analysing the communication difference between the data push versus the data pull approach in this scenario, we find that there is a significant reduction (approximately reduced by a fifth over a sixty second period) achieved when using the data pull. We now assess what differences exist in domain performance between these two approaches.

### Domain performance

Having assessed the communication metrics to understand the variance in data push versus data pull scenarios, we now need to assess how the strategies perform in terms of the problem

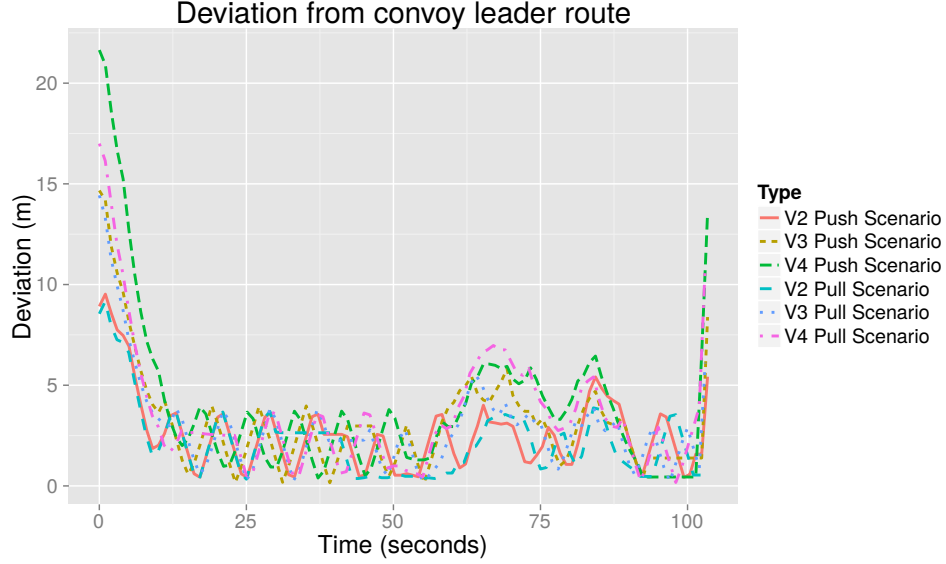


Figure 6-8: Deviation from lead vehicle route

domain.

As there is no road-following model in the `xmppVehicle` simulation of vehicles, we can assess how the convoy members have performed in following the lead convoy vehicle, by measuring their deviation from its route at each time step. Here the interest lies in testing the assumption: if communication is too infrequent there may be increasing divergence from the route of the lead vehicle, where we can consider the route following as not having sufficient resolution (i.e. insufficient data points to follow). Conversely, if the frequency of transmitted data points is high, then the rest of the convoy receive a high resolution route to follow and should show little divergence.

In Figure 6-8 the results of this route deviation analysis are shown. Initially, from the start of the scenario to approximately 13 seconds it can be seen that there is some obvious deviation, as the vehicles are moving from their start locations to begin following the path of the convoy leader. After this period, up until approximately 100 seconds, the vehicles are in steady-state of following the convoy leader's position. For the first period of this 'steady state' behaviour, performance shows a deviation approximately 4 meters or less. However, from approximately 60 seconds to 90 seconds it can be seen that there is some disruption to the earlier pattern of vehicle deviation, occurring as the vehicle moves through the corner section (shown earlier in Figure 5-4) of the route. Comparing like for like vehicle (e.g. V2 push results versus V2 pull results) we can see very similar performance, operating within the same performance envelope and both showing the disruption which occurred at the 60 to 90 second period.

Considering another metric of domain performance, in Figure 6-9 the distance to the convoy leader for each following vehicle is shown. Whilst the earlier Figure 6-8 showed deviation from

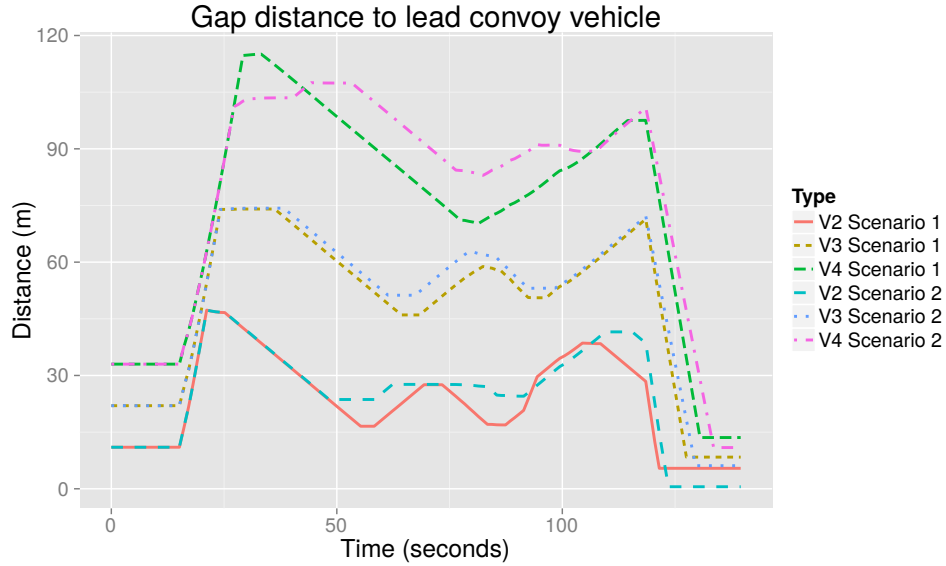


Figure 6-9: Distance to lead convoy vehicle

the route, this would not represent performance in terms of ‘maintaining the gap’ to the vehicle ahead; a factor affected predominantly by the speed control of following vehicles, which is based on their calculated distance to the vehicle ahead.

The distance to leader shown in Figure 6-9 shows a fixed gap until approximately 20 seconds, at which point the leader vehicle sets off, requiring the following vehicles to react and set off at the same speed. However, this gap increases, before starting to decrease (at approximately 25 seconds), which suggests the design of the speed control may lack the granularity to respond appropriately. This can be explained by considering the gap definitions used:

1. If `distanceToCarInfront(0, 10.999999999, applyBrakes)`: If very close to vehicle ahead, then brake.
2. If `distanceToCarInfront(11, 34.999999999, slowDown)`: If more than 11m but less than 35m to lead vehicle, then slow down to increase gap.
3. If `distanceToCarInfront(35, 45.999999999, standardSpeed)`: If between 35m to 45m then maintain speed, gap is OK.
4. If `distanceToCarInfront(46, 200.999999999, speedUp)`: If more than 45m away from leader, speed up to close gap.
5. If `distanceToCarInfront(201, 1000, lostMyLeader)`: If very far from leader, assume have lost contact.

Considering these definitions in relation to the trends visible in Figure 6-9, it can be seen that the distance definitions do not trigger exactly what might be expected, for example vehicle

V2 can be seen to get to approximately 15m close to the lead vehicle on two occasions, where the `slowDown` behaviour could be expected to have been slowing the vehicle, since the gap went below 35m. However, both `speedUp` and `slowDown` operate on a cumulative basis, modifying speed between an upper and lower boundary for the vehicle, for example the `speedUp` behaviour triggers the plan `+!speedUp : standardSpeed(V) & lowerSpeed(L) & upperSpeed(U) & speedInterval(I) & V2=(V+I) & (V2 <= U)`. Here, the speed will be increased by I to a new speed of V2, if V2 is less than the speed limit set for the vehicle of U. This has the impact of modifying the speed over a number of calls of `speedUp`, which prevents over-reaction but may increase the overall reaction time. Here we consider two aspects of this approach where alternatives may offer an improvement. Firstly, that a different approach to maintaining distance may prove more effective; this could take the form of vehicles communicating alternative (i.e. higher level SA) information, or potentially an institution could become involved in an effort to regulate vehicle speeds more closely. Secondly, that this may be a candidate for the tuning capability discussed in the further work of Section 7.1.4.

For the rest of this scenario run, the vehicle gap shows some small fluctuations as vehicles modify their speeds to maintain the distance, although it can be seen at approximately 75 seconds the trend of there being a reasonably constant gap between each vehicle is broken. As in Figure 6-8, where disruption was seen at a similar time, if a vehicle begins to ‘cut the corner’ of the curve in the road then it artificially shortens the gap to the vehicle ahead, placing more strain on the gap-maintenance approach of the speed modification algorithm. Finally, at approximately 120 seconds, the lead vehicle comes to a stop, and the remaining convoy members slow down and brake until they too are stationary behind the leader.

Comparing the results between the data push versus data pull approaches, as with Figure 6-8 it can be seen that the two approaches perform reasonably similarly. Potentially it looks like the data pull scenario may ‘dampen out’ the behaviour slightly, resulting in less over-reaction and modification of speed. For example, if speed is increased in order to reduce a gap, with more frequent exchange of information, this would result in additional calls to `speedUp`, which may cause the issue of closing up too quickly to the vehicle ahead, and having to reduce speed to compensate.

In general terms, it would seem that even with a relatively basic design approach to following the leader, domain performance is satisfactory in both the push and pull scenario variations. Given the more significant reduction in communication overhead, this suggests it is possible (and perhaps beneficial) to communicate less whilst still achieving the same goals.

### 6.3.2 Waypoint exchange

In this scenario, a convoy group of vehicles is established with the goal of travelling a specific route around the city route (as shown earlier in Figure 5-2). In comparison to the previous scenarios, which explored convoy navigation based on the exchange of low level perception information, the waypoint scenario is based on the communication of higher level information. To do this, the convoy leader informs the convoy member vehicle behind of its intended route,



then that vehicle passes it to the convoy vehicle behind it, and so on, until the whole convoy is informed of the route.

As this scenario is intended for comparison to the previous perception based scenario, we first set out some differences and similarities between these approaches. The scenario is significantly longer, taking approximately 10 minutes to travel the entire route compared to the approximately 2 minute duration of the perception based scenarios. This is needed to introduce a more significant number of waypoint locations to the route, and to increase the simulation time such that differences between the two approaches may become more apparent. An analysis was performed of percept updates and message counts, but values were found to be very similar to the data-pull scenario presented earlier in Figure 6-5 and Figure 6-6. Whilst this was expected to be lower, we find that re-use of existing plans is resulting in inefficient communication as vehicles are still communicating their position in order to manage the convoy gaps. However, to remove this communication would require additional sensor libraries to be developed, to allow vehicles to calculate their gap to the vehicle ahead, which raises the prospect of significant development effort to quantify the final reduction in message quantity. Instead, we leave this as potential future work, and assess other metrics of this scenario.

The analysis of route deviation is also omitted here, as in contrast to the perception based scenarios where convoy followers were told x,y positions to move to based on the leader position (and with no road following model, could take shortcuts to get to those positions), here convoy followers receive the exact route the leader is going to take. We consider other metrics which may indicate improved performance compared to the perception based exchange scenarios, and explore measurements of acceleration and braking.

With a convoy approach based on ‘follow the leaders last position’ (as in the perception based scenarios), there is a potential issue that the convoy-follower vehicles may slow down on approaching the desired x,y and then have to speed up again when informed of a new x,y to move to. There may also be changes in their distance to the leader (as shown earlier in Figure 6-9) requiring speed changes to close the gap, the risk of overcompensation resulting in excessive braking, followed by the risk of too much acceleration to then close the gap leading to a cyclical overbrake-overaccelerate-overbrake pattern. Whilst such effects could be mitigated through more sophisticated control strategies, we are interested in the effect of different approaches to communication rather than trying to hide underlying problems.

In seeking to perform a comparison between perception based and waypoint based approaches, we first re-run the perception based approach against the longer route, and capture metrics of braking and acceleration. The results of this are shown in Figure 6-10, with the top graph showing a normalised version of the results shown earlier in Figure 6-8 such that the distance gap is reported to the vehicle ahead (i.e. from vehicle two to vehicle one, vehicle three to vehicle two, and so on). Here we can see all convoy members maintain a similar gap of between 20m to 40m throughout the duration of the scenario, reasonably similar to the results shown earlier in Figure 6-8. In the bottom graph of Figure 6-10, the results of capturing the amount of braking and acceleration are shown, and here it can be seen that acceleration follows a reasonably linear increase up to approximately 150 acceleration occurrences at the end of the

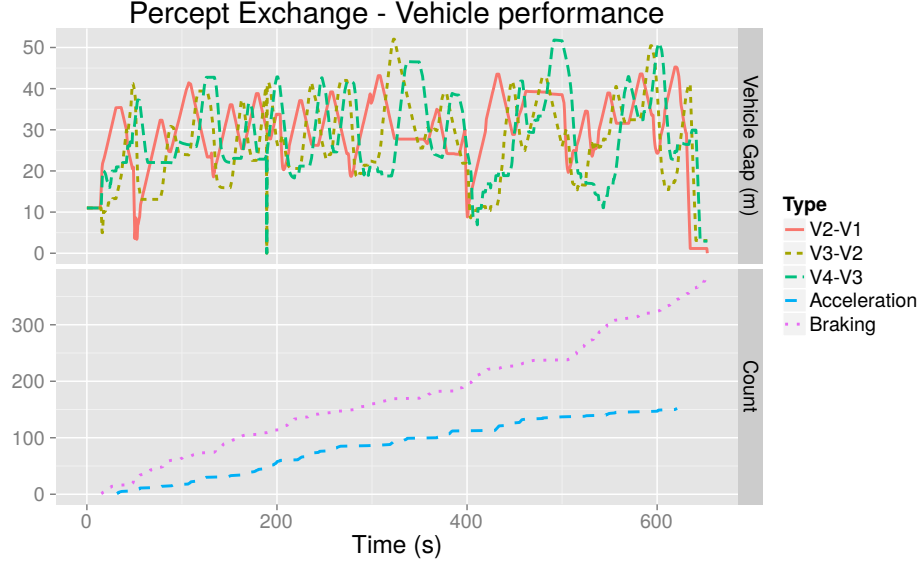


Figure 6-10: Performance of perception based approach

scenario. The amount of braking is almost linear, but there are three cases visible in the lower (Count) graph of Figure 6-10 where there is a sudden increase:

1. At 400 seconds: This is attributed to a reaction to a decrease in vehicle gaps (visible in the upper, Vehicle Gap graph).
2. At 500 seconds: There is insufficient information available to explain this, and additional metrics would be required to obtain suitable information to explain the behaviour.
3. At 600 seconds: This is due to vehicle braking as they become stationary at the end of the scenario.

In Figure 6-11 the results based on waypoint exchange are shown, showing vehicle gaps and the occurrence of braking and acceleration. Apart from an initial spike in the gap of V3 to V2, the gap results show similar characteristics to the results shown earlier in Figure 6-10, with a gap maintained around 20m to 40m. However, the results for acceleration and braking show a significant difference, of approximately half the quantities shown in Figure 6-10 with the amount of braking occurrences at 150, and acceleration occurring approximately 75 times.

This suggests a significant difference between the two approaches, and whilst more sophisticated control and agent strategies may improve the behaviour in perception exchange scenarios, these results suggest that less communication does not result in an adverse effect on the behaviour of the convoy. Considering the problem statements of Section 3.2, this provides an indication that if an appropriate information type is exchanged (i.e. in this case, it is more appropriate to exchange the higher level route information than a constant low level update of positions) then the quantity of information exchange can be reduced.

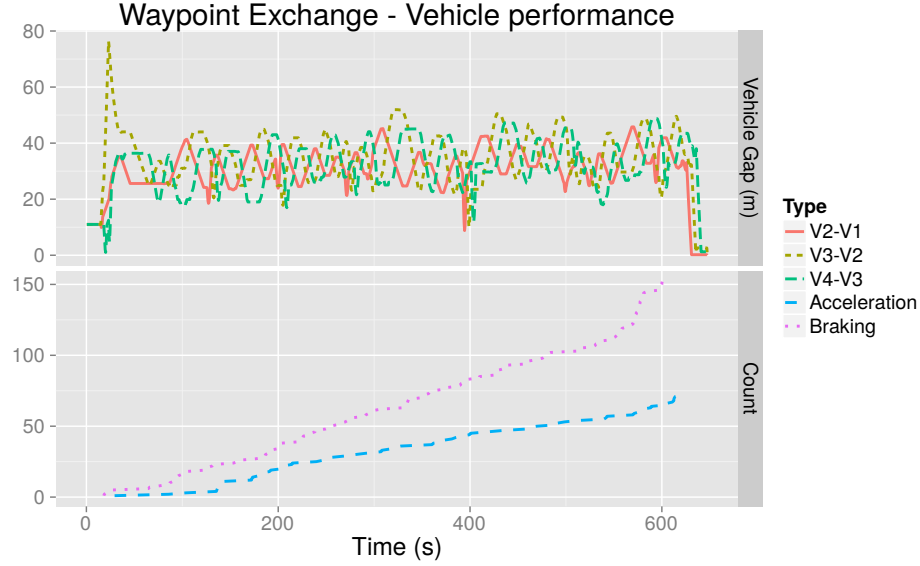


Figure 6-11: Performance of waypoint approach

### 6.3.3 Projection of future events

In this scenario, we focus on the high level projection element of situational awareness, in projecting what impact future states may have in the current environment. There is also the introduction of balancing individual against global desires, as well as touching upon resolving problematic situations. To address these needs we explore the use of an institution as a governance structure, to offer guidance to intelligent vehicles such that the wider collective desires are met.

We revisit the city centre context of the previous scenarios, but with additional features. As set out in the scenario description in Section 5.4, we draw on the concept of traffic lights to provide the time consideration of this scenario, where given a current traffic light state, it can be assumed in some short time period (e.g. 30 seconds) it will change to another state (e.g. from green to red). Since a vehicle knows its own route and speed, it is possible to identify which traffic light(s) will govern the route, at what time the vehicle will arrive at that light, and using this information the vehicle can manage its speed such that it arrives at the light when it is green.

In this scenario, such speed management is enforced by an institution, where in order to prevent a vehicle arriving at a red light, the institution issues an obligation to slow down for a given time period. Since an agent controlling a vehicle may be motivated by other goals (e.g. move as fast as possible to destination), rather than global goals (e.g. reduce fuel and pollution), we see the institution having the role of limiting the agent's autonomy for the collective benefit.

The results of this are shown in Figure 6-12, where two plots of fuel use are presented, one showing instantaneous fuel consumption, and the other cumulative fuel use. Whilst we can

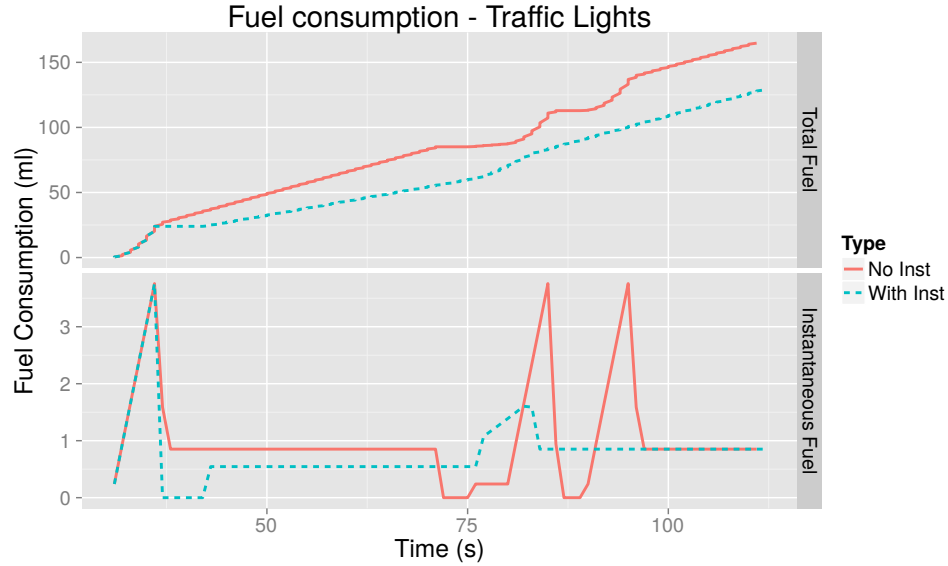


Figure 6-12: Fuel use comparison with and without institution active

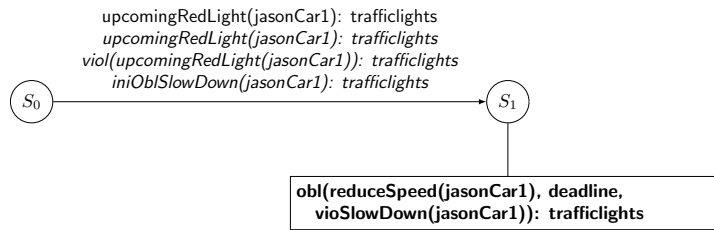


Figure 6-13: Institution states for traffic light management scenario

initially see similar behaviour for the variations of the scenario, with and without an active institution, after approximately 25 seconds the fuel use profiles diverge.

Whilst a narrative is provided here to describe this difference, it is best be accompanied with the provided links to videos generated of the scenarios. For the variation where there is no institution active (video available at <http://dx.doi.org/10.15125/BATH-00080> [Traffic\_No\_Institution.mp4]), the vehicle travels at the appropriate speed for the road, until at approximately 75 seconds when it arrives at the first junction controlled by traffic lights (referring back to Figure 5-5, this is indicated by '1' on the diagram). When stationary at this light, fuel consumption drops to near zero, but when the light changes to green, the acceleration from stationary causes a significant spike in fuel use. The vehicle then encounters another red light at approximately 90 seconds, resulting in fuel used drop off, until it accelerates again when the light turns green. The final traffic light is green, and so the vehicle is able to continue until it reaches the end of the route.

The institution involvement in this scenario was outlined earlier in Section 5.4 (video available at <http://dx.doi.org/10.15125/BATH-00080> [Traffic.With.Institution.mp4]), but in summary it is used in combination with the Area Of Interest (AOI) module which monitors traffic light states along a vehicles route, and produces a message of the form `upcomingRedLight (VehicleName)` if that vehicle is approaching a red light. The institution is subscribed to these AOI messages, which results in the institution state changes shown in Figure 6-13. On receipt of this message, the state  $S_0$  transitions to  $S_1$  due to the observation of `upcomingRedLight (jasonCar1)` (where `jasonCar1` is the agent vehicle name). The resulting obligation `obl (reduceSpeed (jasonCar1))` can be seen at state  $S_1$ , and on receipt of this the `jasonCar1` vehicle complies and reduces speed. The name of the institution is shown in Figure 6-13 as `trafficlights`, as the role of the institution in this scenario is considered to be traffic light centric, where each traffic light could have its own institution, and interact with other traffic light institutions to provide coordination for the vehicle population. Alternative views of institution focus are put forward by other scenarios in this work (revisited in Section 6.4), but for the purpose of managing approaching vehicles, this traffic light centred approach is shown to be effective.

In this scenario, we have explored how the wider society desires (reducing fuel consumption and associated environmental benefit) can be achieved, against the pursuit of individual desire, through the use of an institution issuing appropriate obligations. In the domain of intelligent vehicles, we achieve this by issuing an obligation which ensures the vehicle arrives at a traffic light whilst it is green, rather than arriving whilst it is red (resulting additional fuel to accelerate from stationary when it changes green). The institution has a clear impact, in terms of fuel consumption in Figure 6-12 we see an approximate 20% reduction, whilst the total journey time for the vehicle is slightly increased from 1 minute 44 seconds to 1 minute 50 seconds, an approximate 6% increase. However, this scenario would need a significant increase in both duration and complexity (i.e. additional traffic lights, and representative background traffic) to assert strong domain benefits. This would need to be coupled with further domain specific activity such as tuning both the duration and speed of the `slowDown` period, but in general terms the demonstration of an intelligent agent pursuing externally moderated goals, rather than the sole pursuit of its own, serves to show the application of the concept.

### 6.3.4 Resolving unknown behaviours

In this scenario, we investigate further aspects of the problem statements put forward in Section 3.2, with focus on the resolution of problematic situations. This scenario explores a situation where on a motorway section, a slower vehicle is delaying a faster moving vehicle which wishes to get past. One option for the faster vehicle is to change lanes and overtake, but another approach sometimes adopted is to flash the headlights, to draw the attention of the vehicle ahead and indicate the desire to get past. This technique is discouraged for numerous reasons (e.g. it is perceived to be aggressive, is ambiguous and also unregulated) and the slower vehicle has no obligation to comply. However, we draw on this as an example of social

convention, where although there may be no enforcement mechanism, the social norm exists that some action should be performed. The difficulty with such conventions, is as they are not prescriptive, there is no ‘rule book’ for how they should be carried out, and furthermore they may be culturally dependent (e.g. a flash of headlights could carry a different meaning in different countries) as well as situationally dependent (e.g. on a motorway compared to in a city).

Our hypothesis is that the introduction of institutions can provide a late-binding mechanism to guide appropriately for the social convention of a particular situation. It could be considered akin to taking an etiquette advisor when embarking on interaction with a new culture; it is very challenging to prescribe ahead of time what to do for every possible situation, and some additional guidance may be needed. In the case of this particular scenario of headlight flashing, there is no real world governance in place for such cases, we instead adopt it as an illustrative example of the late-binding notion to provide intelligent agents with assistance. Whilst in this particular scenario both vehicles are controlled by agents, we are representing human to agent communication, considering the general case that where there is mixed human-intelligent agent interaction, difficulties may arise due to ambiguity (of human communication), and complexity (due to cultural and contextual meaning) of a situation.

We refer the ‘indication’ (in this case, the flashing of headlights) to an external body for mediation and issuance of appropriate guidance. If intelligent vehicles are to operate in an environment with human controlled vehicles, there is a need to understand and respond to human indications (such as the use of headlights) and intelligent vehicle interaction (e.g. performed over some wireless communication technology). Whilst the trigger is different in these cases, the mechanism of resolution by an external body stays the same, dealing with the problem of what action is appropriate to bring about a resolution (for example, on the motorway the flash of lights may indicate a desire to overtake, but in a city it may indicate permission to pull out in front of another vehicle). However, the implementation of that action remains under the agents control. The institution only issues an obligation, which might indicate a state (to be achieved) or an event (to bring about) or (at the lowest level of granularity) an action (to take), all of which may be ignored.

We present two variations of this scenario, in the first the flashing of headlights is not resolved and the slower vehicle takes no action. In the second variation, this behaviour is handled via an institution, which results in an obligation being issued to the slower vehicle to change lane.

In Figure 6-14 the comparison between the scenario with and without an active institution can be seen. This is a snapshot of vehicle speeds and gaps when the scenario is at 59.5 seconds, a point at which without an institution the vehicle wishing to overtake has had to perform a hard brake, to avoid colliding with the slower vehicle. Because of this speed reduction, the most noticeable difference can be seen in the bottom graph, where at approximately 1250m along the route, a localised disruption can be seen. One vehicle can be seen to be travelling around 25mph, with the vehicle slightly further along the route (i.e. the vehicle ahead of it) being the slower moving ‘lead’ vehicle which has not acknowledge the flashing headlights. Behind the

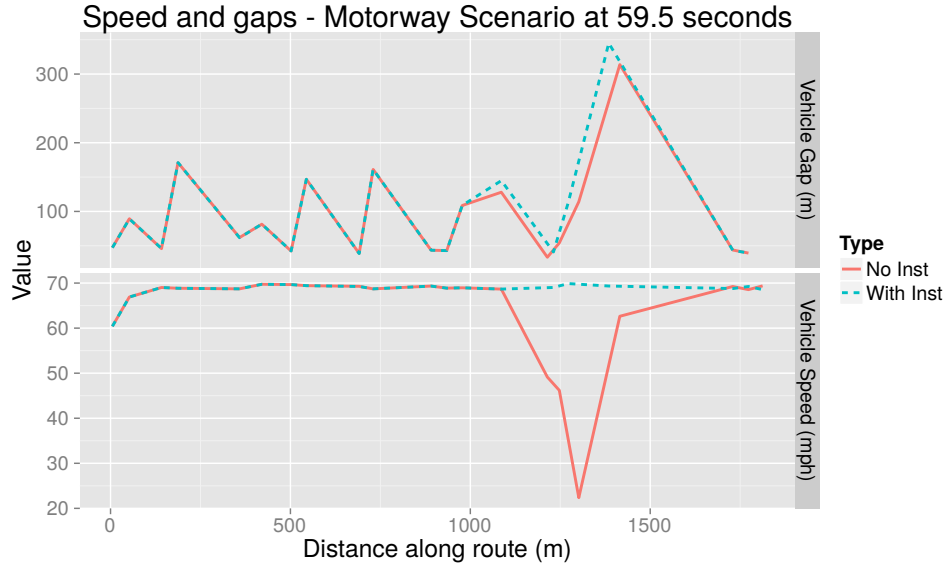


Figure 6-14: Vehicle speed and gap comparison for individual lane

vehicle travelling at 20mph, it can be seen that at least one other vehicle has been affected by the hard braking, where it has had to slow down to approximately 45mph.

By comparison, Figure 6-14 show the impact of the institution. Here, nearly all vehicles are travelling at 70mph, with no disruption caused. The flash of headlights triggers the institution to issue an obligation for the slower vehicle to change lane, resulting in the obstructed vehicle not having to slow down. It can be seen in the non-institution variation of this scenario that there is a vehicle travelling at approximately 60mph nearly 1400m along the route (and this is the vehicle causing the delay). By comparison, there is no plot point at this value in the variation with the institution active, as that vehicle received the obligation to change lane, and so as Figure 6-14 is only showing vehicles for one lane, the plot point is no longer shown.

The scenarios can be reviewed along with video recorded from a number of the components, which provides a view of how the behaviours develop in realtime. For this non-institution variant, this is available at <http://dx.doi.org/10.15125/BATH-00080> [FlashLights\_No\_Institution.mp4], and at the <http://dx.doi.org/10.15125/BATH-00080> [FlashLights\_With\_Institution.mp4] with the institution active.

The institution state change is shown in Figure 6-15, transitioning from state  $S_0$  to  $S$  with the observation `flashLights(jasonCar1)` where `jasonCar1` is the slower moving vehicle (which the flash is targeted at). This results in the issuance of the obligation `obl(changeLane(jasonCar1))`, at which point the slower moving vehicle changes lane, allowing the faster vehicle past. Here the institution's role is considered more 'vehicle-centric' (as in Figure 6-15 the institution is referred to as 'roadusers'), reflecting how the larger collective of road users could benefit from individually targeted obligations to aid the wider population.

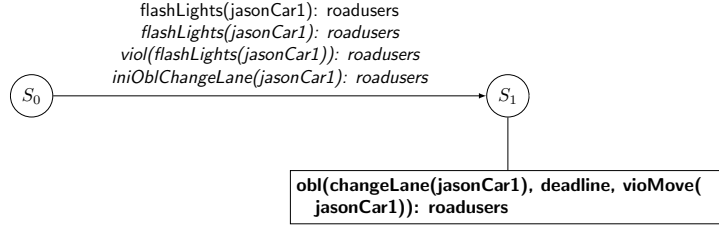


Figure 6-15: Institution states for ‘flash lights’ resolution

In this scenario we have shown the use of an institution in providing assistance to an intelligent agent for a case where that agent has no appropriate knowledge itself of what to do (and, is unaware it needs to take any action). We highlight the need for this capability through the scenario of one vehicle flashing its headlights at another as an indication that it wishes to get past, but we use this to illustrate a type of situation where social convention (including cultural and contextual issues) gives rise to a difficult situation for an intelligent agent to handle. Without the institution, the obstructed vehicle is hindered, resulting in erratic behaviour of braking followed by acceleration. With the institution active, the desired meaning of the flashing headlights is translated into an action for the slower moving vehicle, which when obeyed results in less disruptive behaviour and negative impact on the rest of the traffic flow, and demonstrates an example of our intended late-binding mechanism of institution assistance for intelligent agents.

### 6.3.5 Variable speed limit

In this scenario, we focus on the problem statement from Section 3.2 concerned with balancing individual pursuits versus the greater social welfare, where a number of individual vehicles are required to travel at a significantly reduced speed (contrary to their desire) in order to reduce congestion (for the wider society benefit). We demonstrate the use of an institution in a governance role in this scenario, where its use is triggered by a vehicle braking hard, at which point the institution issues obligations to a number of affected vehicles in order to coordinate a response to dampen any congestion pattern that may have arisen due to the braking behaviour.

In the Variable Speed Limit (VSL) scenario outlined in Section 5.6, a single lane of traffic within the motorway context is used to investigate how the impact of the vehicle suddenly braking hard can be mitigated through enforcing a speed limit, issued either globally (similar to the current process used on real roads) or via an institution issuing obligations to specific vehicles. This provides a means to contrast the institution approach to the global approach, but it presents challenges in extracting appropriate domain specific metrics to establish the impact of the proposed institution based approach.

With this approach, there is interest both in how the whole population of vehicles is affected, as well as the specific difference in behaviour of the intelligent agent vehicles. For this reason, we





Figure 6-16: Agent controlled speed comparison for VSL scenarios

first analyse the speed behaviour of the intelligent vehicles, comprising of C1 (the lead vehicle which performs the brake), C2, C3, C4 (the three which receive institution obligations to slow down) and C5 which is not issued any obligations in this configuration.

In Figure 6-16 the speeds of these five vehicles over the course of the scenario are shown, for the variations of no VSL, an institution controlled VSL, and a global VSL. In all three variations, it can be seen that the lead vehicle (C1) performs the hard brake at 110 seconds, coming to a near stand still before resuming to the appropriate speed (of approximately 38m/s where there is no VSL or institution VSL, and up to approximately 25m/s when affected by the global VSL). The main interest lies in the behaviour of vehicle C2-C5, and examining the no VSL variation first, it can be seen that these vehicles show a slow decline in speed from C1 braking at 110 seconds to approximately 125 seconds. After this, their speeds loosely follow that of C1 and return to normal velocity at approximately 190 seconds. Similar behaviour can be seen in the global VSL variation, although one notable difference is all vehicles reduce their speed at 110 seconds to comply with the the new speed limit. This allows vehicles C2-C5 more time to react, and their speed reduction can be seen to be more gradual, followed by a recovery to match the speed of C1 at approximately 150 seconds.

The variation of behaviour with institution enforced speed limits in Figure 6-16 proves more challenging to interpret. Firstly, it can be seen that the institution obligation to reduce speed is obeyed, as vehicles C2-C5 decrease their speeds according to the medium, mediumSlow and slow modifiers received.

#### Listing 6.1: Agent reaction to slowDown event

```
+slowDown(Value) : speedModifier(Value, NewSpeed) & driverAgentName(DAgentName) <-
```

```

.print("asked to slowdown to ", Value, " which I translate to ", NewSpeed);
.send(DAgentName, achieve, setSpeed(NewSpeed));
.wait(5000);
.send(DAgentName, achieve, setSpeed(-1)).

```

---

The handling of the received a `slowDown` event is shown in Listing 6.1, which explains the observed behaviour, as the driver agent implements the reduced speed, waits 5 seconds, and then returns to automatic speed control (indicated by the -1 value). With this obligation removed, in Figure 6-16 these vehicles are shown to accelerate until at 125 seconds, at which point they slow down again. This time the speed reduction is due to getting too close to C1, and the speeds at 150 seconds are very similar to those in the no VSL scenario version at the same time point. However, between 150 seconds and 200 seconds the vehicles show a different speed profile compared to the non VSL scenario, seemingly taking longer to increase their speed. Whilst the two sets of behaviour (accelerating too much, and then not enough) are not intentional, we presume there is some feature of the simulation configuration causing these (e.g. driver reaction times). Whilst these could be the subject of future work to fully understand the behaviour (as adding the visibility of such SUMO reasoning would help the general problem state of observability), the more general issue of how an agent returns to its previous behaviour after institution intervention is of significant interest, which we revisit in the wider time considerations of institution behaviour discussed in the further work of Section 7.3.3. Considering the behaviour shown in Figure 6-16, some additional management in the 75 second region of the simulation, and after 150 seconds, might help enforce a more stable agent performance. A more general consideration of such a requirement follows in Section 6.4.3, considering how the limitation of autonomy needs to be followed with a process to add it back, over a period of time.

However, whilst this analysis shows the impact of the three variations of the scenario in terms of vehicle speed, there is the question of how it has impacted the wider vehicle population. Here we draw on the motorway metrics put forward earlier in Section 6.2.4 and review all vehicle speeds along the route in combination with the distance to the vehicle in front, on the premise that any location where there are groupings of vehicles at low speed and small gap to distance ahead, is an indication of potential congestion.

In Figure 6-17 the results of vehicle speeds and gap distance ahead are shown for the three variations of VSL approach, taken at 240 seconds into the scenario, when the simulation ends. The interpretation of these results proves more challenging, as this is a single ‘snapshot’ of the simulation state, where any disruptions could be limited to just that time interval (e.g. a vehicle braking hard). It is in these situations where reviewing the ‘live’ unfolding of events through the various BSF components provides a better understanding of what is taking place, and four key component outputs have been captured and combined into a series of videos for this purpose. These are available online: the no VSL version available at <http://dx.doi.org/10.15125/BATH-00080> [VSL\_No\_Institution.mp4], the institution VSL version at <http://dx.doi.org/10.15125/BATH-00080> [VSL\_With\_Institution.mp4] and the global institution version available at <http://dx.doi.org/10.15125/BATH-00080> [VSL\_GlobalSpeed.mp4].

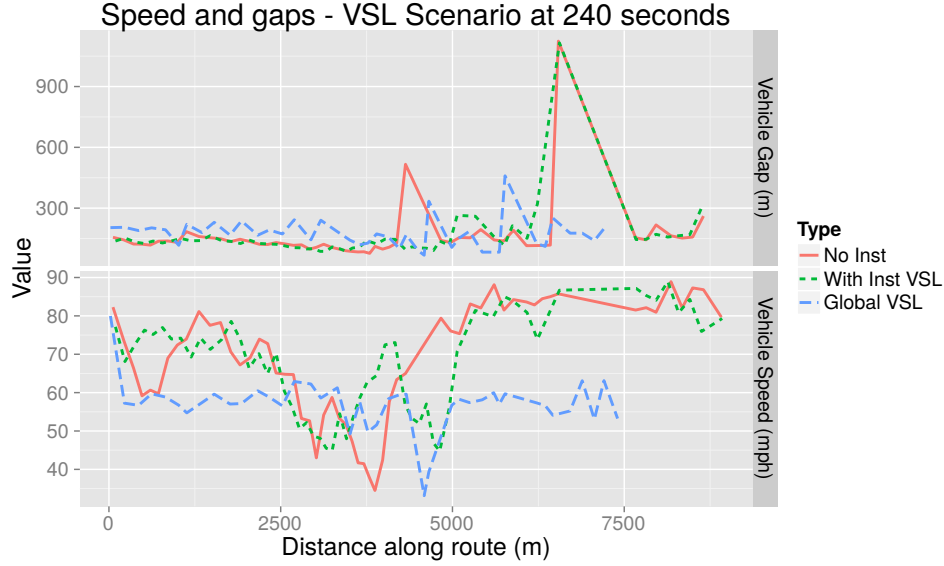


Figure 6-17: Vehicle speed and gap comparison for VSL scenarios

Considering these videos, and focussing on the results shown in Figure 6-17, we can perform some further analysis. Firstly, we can identify the position of the lead vehicle which performed the hard braking (initially done at 110 seconds), as there would be a large gap to the vehicle ahead of this (caused during the slowdown while the vehicle ahead carried on at its normal speed). This can be seen at approximately 6250m along the route, where vehicle speeds have returned to normal. Considering the speed of C1 (the braking vehicle) shown in Figure 6-16, it can be seen C1 recovers to 38m/s eventually, though for a period of this time is accelerating from 0m/s. Therefore, for an approximate speed of 20m/s from the 110s brake point to 240s end of scenario, the vehicle would have travelled somewhere in the region of 2600m. Of interest is that at this distance back along the route (approximately 2625m) there is an area of vehicles travelling at low speed for the no VSL scenario. We consider that there may be a static congestion wave formed at this location, caused by the initial heavy braking, which when left undamped, remains, as new vehicles arrive at the back end of the wave, and vehicles leave the front to accelerate back to the normal speed.

Comparing the institution VSL results of Figure 6-17, although there are still some areas of reduced speed, they seem to have been dampened compared to the non institution approach, decreasing to 45mph rather than 35mph. At this area of reduced speed (approximately 2625m along the route) the vehicle gaps also show less reduction when using the institution approach. Given that the approach of the institution has been left largely ‘untuned’ i.e. affects only a small number of vehicles, and reduces the speed to somewhat arbitrary values, this seems a significant difference. This presents an area where further tuning (e.g. of geographical scope of the institution, how much to reduce speed to, and for what duration) could improve the result

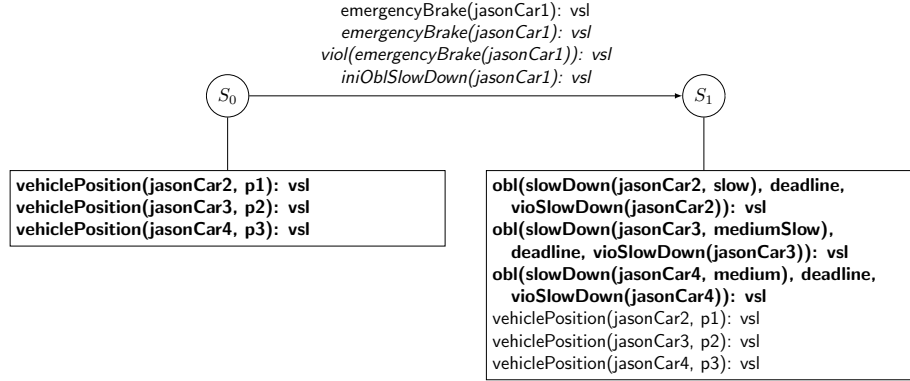


Figure 6-18: Institution states for VSL scenario

of this approach.

Considering the global VSL approach results of Figure 6-17, it is difficult to draw many strong conclusions. There are artefacts of the slower speed, visible directly by the speed reducing to approximately 55mph along the route, and also that less distance is travelled during the 240 second simulation compared to the other two scenario variations. However, an area of reduced speeds can be seen at approximately 4000m along the route, quite similar in profile to the reduced speeds of the no VSL scenario variation. Reviewing the video (<http://dx.doi.org/10.15125/BATH-00080> [VSL\_GlobalSpeed.mp4]) of how this has evolved, the vehicle speeds are all reasonably constant at the global VSL value of 50mph, until around 210 seconds into the scenario when an area of slower vehicles starts to develop. Whilst interesting, this is another area which we leave for potential future work, as such analysis pulls us deeper into the problem domain itself when the institution approach has clearly demonstrated an effect where we feel additional tuning could improve domain performance, but our interest lies in highlighting the generic capability.

We draw on the trace shown in Figure 6-18 to highlight the transition from an initial state where there are a number of vehicles behind `jasonCar1`, the event of `jasonCar1` braking, and the resulting institution guidance. At the initial  $S_0$  the institution is informed of the current vehicle positions behind the braking vehicle (e.g. `jasonCar2` is the first vehicle behind in `p1`), and the observation `emergencyBrake(jasonCar1)` during the transition to state  $S_1$ . At this point, a number of obligations can be seen at  $S_1$ , where the vehicle positions are unified to an appropriate speed modification (e.g. `p1` results in `slow`). Here the institution is providing a coordination role, and compromising a smaller number of individual desires (vehicles which would want to go as fast as possible) in an attempt to resolve the congestion wave for the benefit of the rest of the population.

The configuration of the institution enforced VSL is an area where improvements could be made; parameters such as how much each following vehicle should have to slow down, and for how long, could be viewed as an optimisation problem. This raises a benefit of the simulation

based approach, as repeat runs can be performed while such parameters are changed, with the impact of such modifications assessed to inform the parameter choice of the next simulation iteration. There is also a set of parameters requiring tuning to identify the optimum number of vehicles which should received VSL obligations; in this experimentation we chose the number to be three, but it could be more than this (assuming the vehicles are intelligent agent controlled rather than human controlled), or alternatively it could be calculated on a geographical basis (i.e. what vehicles are with a certain radius of the braking vehicle).

This raises the topic, relating back to the problem area of introducing new technology, as to how vehicles controlled by human drivers could be included in institution management. For example, if it is found that localised VSL control targeted by an institution to specific vehicles could alleviate the need for global limits, it may be desirable to pursue this further and tackle the issue of how humans drivers would be included. Although such general topics of the work are revisited later in Section 6.4, we highlight it here as specific to a successful implementation of such targeted speed limits, where there is a need for all vehicle types to comply.

### 6.3.6 Post-accident management

In this scenario, we explore a situation where there may be involvement of multiple bodies, where they hold jurisdiction over different aspects of the scenario, in order to investigate the problem statement from Section 3.2 of resolving difficult situations. This also extends the consideration of balancing individual goals against the greater collective, demonstrating the ability of institutions to interact in resolving a situation.

A situation where a crash has occurred on a motorway section provides the scenario, involving two parties where an intelligent vehicle (referred to as `jasonCar1`) has driven in the back of another vehicle (referred to as `victim2`) and is no longer able to move. On becoming stationary in its motorway lane, this causes an obstacle for the vehicles behind, where normally they have to attempt to change lanes in order to get past and proceed along their route. However, this is a case where the imperfect nature of vehicle communication becomes a problem, as whilst the blocked vehicle can transmit its desire to change lane through the use of an indicator light, there is no formal method for a vehicle in that lane to transmit back a ‘OK / Acknowledged’ message. As discussed in the previous scenario of Section 6.3.5, the use of flashing headlights has multiple uses, and this is another such case, where the desire to change lane can be acknowledge by a flash of lights. In this sense, the scenario also further adds exploration to the problem statement of resolving unknown behaviours, as whilst the desire to change lane is shown through the use of an indicator, there is insufficient information (e.g. when the change lane is planned) to allow other drivers to plan accordingly. This also links back to the need for sufficient information to be included in message exchange, as the short piece of information of an indicator light does not contain sufficient supplemental information to improve other drivers understanding of events. In the SA sense, there is insufficient information to form the projection element of understanding; the vehicle may pull out sometime but it is not known when.

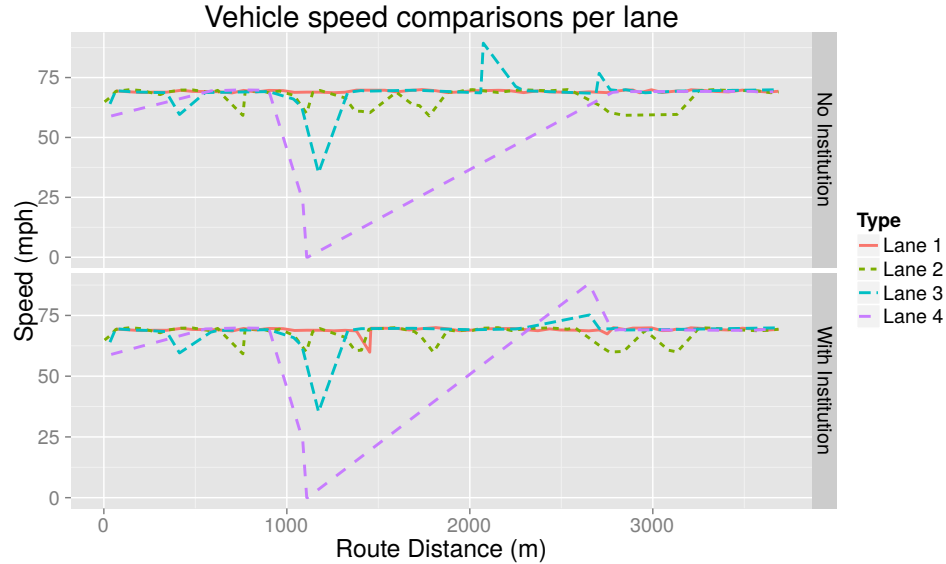


Figure 6-19: Speed comparison per lane for all vehicles

There are two variations of this scenario presented, the first where no institutions are involved, and vehicles have to coordinate their own change lane behaviour when they believe there is a suitable gap. In the second variation, an institution is informed of the crash and (amongst other activities) issues merge pairings, which comprise of a vehicle needing to change lane paired with a vehicle in that lane with the responsibility to allow sufficient space for the vehicle to move into.

In Figure 6-19 the speeds for all vehicles across four lanes of the motorway route are shown. The leftmost lane is 'Lane 4' which contains the crashed vehicle, with 'Lane 3' being the lane vehicles need to move into in order to navigate around the crashed vehicle. The additional 'Lanes 1' and 'Lane 2' are included as there may be some disruption seen there if vehicles move from 'Lane 3' into those lanes. Whilst analysis of these two scenario variations is provided here, this is based around a single timeslice of the simulation, which is best supplemented by videos created of the entire simulation, for the non-institution version available here <http://dx.doi.org/10.15125/BATH-00080> [Crash\_No\_Institution.mp4] and with active institutions here <http://dx.doi.org/10.15125/BATH-00080> [Crash\_With\_Institution.mp4].

In both scenario variations the overall profiles shown in Figure 6-19 are quite similar, with the largest immediate deviation from normal road speed (in both cases) being the crashed vehicle shown with a speed of 0mph, approximately 1100m along the route. Apart from this, there are some other small deviations, but as the results are similar we focus instead on the intelligent agent vehicles to assess the affect of the institution approach.

In Figure 6-20 the speed behaviour over time of six intelligent vehicles is shown. Firstly, there are two general characteristics to draw on, that agent vehicles would travel at a speed of

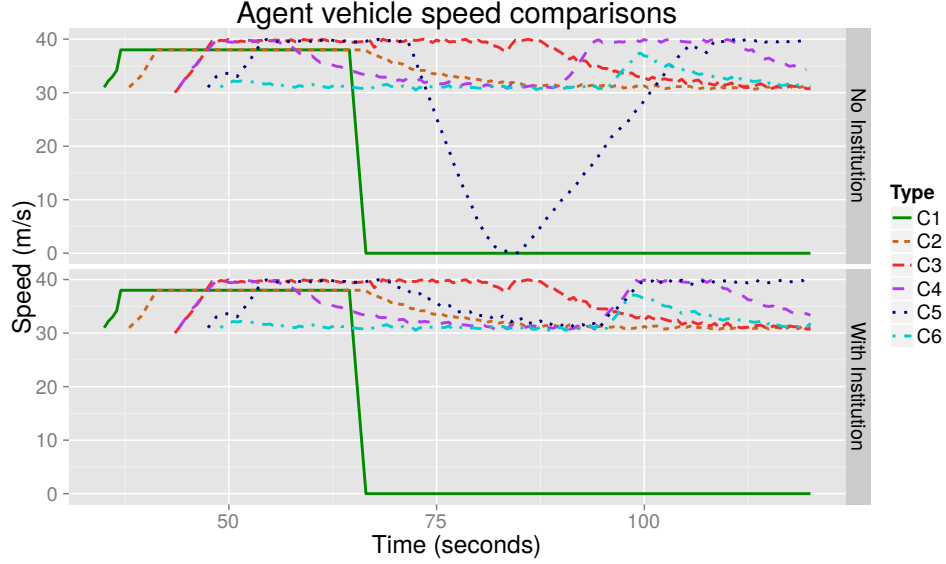


Figure 6-20: Agent vehicle speeds

38m/s if not blocked by a slower moving vehicle, whilst the remainder of the vehicle population would travel at speed in the region of 30m/s (depending on various vehicle types such as lorries, vans, and normal cars). For this reason, in Figure 6-20 most speeds are in a window between 30m/s to 38m/s with gradual transitions between, when either slowed by another vehicle or with a clear road ahead. Outside of this speed behaviour, there are two clear deviations, the first affecting vehicle C1 in both scenario variations, where it comes to a stop after crashing at approximately 65 seconds into the simulation. The second difference relates to vehicle C5, where in the no institution variation of the scenario, C5 slowly decreases its speed before becoming stationary for a few seconds, after which it begins accelerating again. In this case, vehicle C5 approached the crashed vehicle C1, but was unable to change lane due to traffic (in the desired lane), and so eventually had to come to a stop until it was able to pull into lane three, and continue its journey. By comparison, in the scenario with the institution active, C5 has a merge coordinated on its behalf via the institution, with vehicle C6 (as can be seen in the institution diagram of Figure 6-21) which results in C5 being able to continue at normal speed. In more general terms, this shows the application of institutions to form dynamic agreements between entities in order to coordinate a resolution where some individuals profit greatly, some take a small penalty, but the overall society benefits.

In Figure 6-21 the states of the institutions are represented, showing additional institution activity beyond the merge coordination apparent in the results of Figure 6-19 and Figure 6-20, which draws on multiple institution interaction. Considering the state at  $S_0$ , there is a `carInsuranceState` initial state where `jasonCar1` has no valid insurance at the time of the crash. There are also two `mergePairing`'s established, instigated by the crashed vehicle



Figure 6-21: Institution states for crash management scenario

(these could be determined by the Area Of Interest module but are hardcoded in the current implementation). Significant activity can be seen in the transition from  $S_0$  to  $S_1$ , firstly with the `crashed(jasonCar1, victim2)` observation being relayed to all the institutions. Following this, there are a number of institution specific events, `checkInsurance(jasonCar1)` as this vehicle caused the crash and may have to pay, followed by a `crimeReport` as it has no valid insurance. There is also generation of events related to the request of a merge, and the requirement to perform a merge. Finally, at state  $S_1$  there a number of resulting outcomes: the `emergencyServices` institution contains the fluent that there has been a crime committed by `jasonCar1` for having no insurance, and also contains the obligation to dispatch a police unit to the scene of the accident. The `motorwaymanagement` institution issues merge obligations to `jasonCar4` and `jasonCar6` to allow vehicles (`jasonCar3` and `jasonCar5`) to merge into their lane, which results in the improved speed behaviour shown earlier in Figure 6-20. In general terms, we highlight how there has been substantial more activity in addition to the immediately observable (of the merge behaviour), where multiple institutions interact to tackle a wider set of issues such as judicial (was there a crime committed), financial (who is liable to pay who) and logistical (if any emergency services need to be dispatched).

This scenario demonstrates the interaction of multiple institutions, with a demonstrable benefit in the problem domain of improved merge lane coordination. Given the desire to



demonstrate general application rather than domain specific behaviours, the question arises of much to develop this further, for example, a police vehicle could have been injected into the simulation to arrive at the crash location, following the `dispatchUnit` obligation from the `emergencyServices` institution. Consideration is needed as to the benefit of such development, unless there was a planned evolution of the work in that direction. As the mechanism of multiple institutions is shown, interacting to bring about the resolution of some situation, for the more generalised problem level this shows the concept working, without being drawn into implementing further problem domain specific capability. However, there are elements of this scenario where more problem domain development could be useful. The SUMO simulator is capable of representing various traffic scenarios, but has a number of areas which are less mature, such as vehicle crashes, bad driving behaviour, and so on. Whilst the SUMO software is still actively developed, work-arounds and best-guesses have had to be used in this research to show desired behaviour. For demonstration of domain benefits, it would be beneficial in further work to validate models and parameters used in order to replace such customisations with more credible representations, but this draws focus onto the domain rather than the broader set of problems set out in Section 3.2.

As scenarios have developed in complexity, the metrics used have also required further customisation, and this has proved challenging particularly in this crash scenario, where complex traffic patterns emerge and disappear over time. Whilst sufficient evidence has been captured that the institution approach has an impact, the ability to capture behaviour over time in sufficient detail is a likely requirement if further development of this type of problem domain scenario is undertaken. However, the existing capability has allowed us to explore the problem statements of Section 3.2 and gain insight into the suitability of our proposed approach, and problem areas with potential to follow up in further work (discussed in Chapter 7).

## 6.4 General findings

The aim of this work is to investigate the challenges involved in socio-cognitive systems, with a number of specific problem areas outlined earlier in Section 3.2 which provide specific areas for exploration. A system has been developed with components aimed at addressing these problems in a generalised fashion, whilst being grounded in the domain of intelligent vehicles. As such, we need to consider findings from a number of angles, and primarily we break these down into two groups: domain-specific findings, and findings related to the performance against the general problem areas.

In the analysis per scenario presented in Section 6.3, findings from each scenario were presented, with focus on measurements obtained from the problem domain of intelligent vehicles. We now consider these experiments and their findings against the general problem areas put forward in Section 3.1, though a more generalised review against the problems and objectives outlined in Chapter 1 follows later in Chapter 8.

### 6.4.1 Knowledge and communication

A common theme amongst all of the scenarios presented in this chapter is knowledge and understanding, with communication at appropriate levels (considered in the SA context) investigated as means to reduce the quantity of information exchanged.

The perception exchange scenario results of Section 6.3.1 show that just a blind ‘fire and forget’ approach of publishing all possible information can work, but risks overloading both the infrastructure and the agent, with the preliminary findings in Section 6.1 showing issues can occur with an overly large agent belief base, and plan failure can occur if the rate of message exchange is too high.

The results of Section 6.3.1 show that for one of the experimental scenarios, switching from a data-push to a data-pull could reduce agent messages by 90% (whilst still achieving satisfactory domain performance), in Section 6.3.2 the results based on communicating at a different level of knowledge show that a similarly low quantity of messages can be exchanged as the data-pull scenario, whilst still achieving the domain goal of convoy behaviour. However, it also shows that behaviour can be improved, as comparing the results based on high volume but low SA (Figure 6-10) against low volume but high SA (Figure 6-11) it can be seen that the braking and acceleration behaviour has improved when using the higher level SA waypoint information. Whilst there is no specific additional reasoning taking place in the agents about the waypoint messages (i.e. there is no higher level behaviour directly due to the message content), it is providing less interruption to the agent of constant speed and route modifications (based on a stream of position updates from the vehicle ahead). This would seem analogous to advice given to human drivers regarding motorway use: to observe what is happening a few vehicles ahead and make (fewer) gradual corrections based on this, rather than focussing on the vehicle immediately ahead and making a greater number of more extreme corrections.

The benefits of exchanging high level, rich information (introducing the projection element of SA) were explored in Section 6.3.2, finding that advanced planning could take place when the impact of future environmental states was used for reasoning. In this case, it could be argued that representing information at a higher level has allowed various components to communicate some aspect of their behaviour where the summation of these pieces of knowledge has resulted in a very useful, high level message back to the vehicle. The Area of Interest (AOI) module was able to draw on the knowledge of the vehicles route, traffic lights along this route, and given the vehicle’s speed, identify future light states that would impact the vehicle. We also observe that such a capability introduces the concept that in an entirely driverless vehicle world, traffic lights may become superfluous and could be replaced with an alternative coordination structure, although they will be likely to be required as long as human drivers remain in the scenario. In broader terms, such distributed reasoning provides an interesting capability, where if individual components publish suitable information (e.g. route information, speed) other components can provide them with potentially relevant knowledge in return. This links to the issue already introduced in this section, that blindly publishing all possible data may not be an appropriate technique, and so raises the question of how to decide what to publish, and how

frequently. Whilst the idea of publishing to different streams (e.g. at different frequencies) was introduced in Section 3.4.1, there is still the question of what to publish; components cannot overly dedicate resources to constructing every conceivable piece of knowledge, especially if there is no requirement for it. This question is discussed further in Chapter 7 as further work, where we suggest the need for a mechanism to allow one component to request another component to start (or stop) sending data of different types.

An approach considering situational awareness has been adopted across all the experiments, with various levels of information exchange taking place, with the first set of scenarios focussing on whether performance is positively or negatively affected by alternative communication strategies. These show that for different contexts it is possible to reduce the quantity of information exchange and achieve the same performance level (i.e. in these experiments, vehicle convoy behaviour). However, it is difficult to state that the intelligent agents' understanding has improved, although this could be argued depending on what is meant by understanding. As this work draws on the area of SA considerably, then by exchanging higher level information (in the SA sense), an intelligent agents' understanding of the environment (in SA terms) is also improved. Broadening out from the SA view, if an intelligent agent is provided with semantically annotated data, where messages are more advanced than simple sensor readings, and instead reflect more reasoned knowledge, then it could be argued this has the potential to increase agent understanding, both through the message itself and also via the semantic layer (i.e. to further reason about the message).

The artificial system needs to perceive and comprehend actions from humans in the environment, and humans have similar requirements; to understand what the artificial system is doing but to also have this communicated in such a way that they feel comfortable operating in a mixed human-artificial system environment. Such understanding can be considered in the SA context, where we want to develop not just a basic perception of what another being or system is doing, but the comprehension and projection elements in order become more seamlessly integrated.

## 6.4.2 Introduction of novel technologies

A number of potentially negative issues of introducing novel technologies were discussed earlier in Section 3.1.2, and whilst the scenarios presented here do not address these directly, a framework is put forward and demonstrated in the experimental work where future intelligent vehicles can be configured and simulated in conjunction with a representation of existing vehicles.

This integration effort has shown how an intelligent agent layer can be coupled with a simulated environment, allowing scenarios to be constructed and used to explore specific types of interaction between potential technology (in this case, intelligent vehicles) and the current environment. Potential benefits of such future technology were explored in the area of traffic light coordination (in Section 6.3.3), variable speed limits (in Section 6.3.5) and in post-accident management (Section 6.3.6). These scenarios make use of additional capability that the future technology would bring (in the problem domain, vehicle to vehicle communication coupled

with an intelligent agent and governance framework) and demonstrate how this technology could interact, as well as potential benefits such as fuel saving and reduced emissions.

In the scenario focussing on variable speed limits, the results of Figure 6-16 and Figure 6-17 show that the obligations issued by the institution to vehicles to slow down are obeyed, but that their impact is difficult to determine, and that future work could focus on tuning various aspects of the obligation to improve its effect. Whilst this effort is not explored further in order to avoid becoming too domain focussed, the capability of being able to run multiple simulations, in the same conditions, and observe the affect of any parameter tuning, could be a technique of benefit to the problem domain. In broad terms, this could be used to investigate legislation pertaining to future technology, before the technology is deployed in the real world. There is also a requirement for such novel technology to gain public acceptance, and an understanding of what the interaction between existing users and the new technology will look like. We note in a recent funding opportunity<sup>2</sup> regarding research into driverless cars on UK roads that proposals were expected to contain (amongst other requirements) consideration of “interaction with other road users” and “research and investigation of public acceptance”. This would be a potential direction to develop the intelligent transportation domain aspect of this work further (discussed later in Chapter 7), but in general terms shows the potential application of the framework adopted in this work to explore interaction between new technology and existing systems in acting as a bridge between a simulation of the new technology and humans performing their current activities.

### 6.4.3 Ability of intelligent systems

In all of the scenarios presented in this chapter, intelligent agents are at the forefront of the experimentation, and in a number of scenarios (e.g. traffic light coordination and variable speed limits) their performance is contrasted against a *representation* of human behaviour.

The findings have been positive, although there has not been significant strain placed on their deliberative reasoning and intelligence capabilities, instead placing more focus on exposing their mind state and reasoning process and coupling with an institutional framework. The introduction of the institutional model provides the opportunity to off-load agent programming and knowledge codification to an alternative location, and whilst effort is still required to encode such content into an institution definition, the mechanism of distributed management and coordination may be better suited there, than written into each agent. For example, revisiting the convoy scenarios of Section 6.3.1 and Section 6.3.2, there is a governance role required to handle the management of the convoy, for cases such as reducing the convoy speed if a vehicle becomes lost, or creating a gap in the convoy for a new vehicle to move into. Such behaviour could be achieved via agent to agent communication, but as the complexity of the scenario increases it becomes harder to see how this would be managed at an agent level, for example if two convoys need to merge, which agents should be involved, and how would they coordinate the move.

---

<sup>2</sup>Introducing driverless cars to UK roads, <https://www.innovateuk.org/-/introducing-driverless-cars-to-uk-roads#>, accessed 10th September 2014

We highlight that the institution may also offer a benefit in the management of temporal aspects of a problem, as first introduced in the variable speed limit scenario discussed in Section 6.3.5. In this scenario, the vehicle behaviour was affected by an obligation to slow down, and after some time period this obligation expires. Having shifted the vehicle from its previous steady-state behaviour for a period of time, there is a further behavioural requirement to manage the vehicle’s return to its previous state. This can be considered from the perspective of autonomy as well, that the obligation overruled the agents autonomy for that behaviour, and when the obligation expires there may be a need to manage the handing of autonomy back to the agent. For the domain example, if all vehicles accelerate at their maximum rate to the speed limit once the obligation expires, it may create a new congestion wave, and so highlights the need for a managed return to a stable steady-state. Both determining the duration of an obligation, the returning of autonomy, and the management of returning to prior behaviour, seem suited to residing outside the individual agent, as the observability of desired behaviour, and the capability to bring about the required action (e.g. all vehicles slow down to ease congestion) sits not an individual level, but at a collective level, and requires the coordination of that collective to achieve it.

Once the initial problems affecting the agent layer (of managing the belief base, discussed in Section 6.1) were resolved, no significant issues were encountered with the ability of the intelligent agents when used in the scenarios. However, the management of the belief base introduces temporal element, regarding whether beliefs should expire, be overwritten with new beliefs, and so on. The topic of belief revision has been discussed specific to the Jason architecture in Alechina et al. [2006], but we face a larger problem, of what information can be discarded, and what should be kept, as needing to be addressed. In the early scenarios, a vehicle’s belief regarding its current position should be unique as the vehicle can only occupy one position. Based on this, the prior position can be deleted, but, should it be maintained as an expired belief for future reference? If a vehicle is heading to some destination via route A, but then due to traffic switches to route B, what happens if the prior route and traffic congestion knowledge is removed? The vehicle may decide to switch back to route A, and encounter the congestion after all. In more abstract terms, this brings us back to the issue of communication and the quantity of knowledge, being both exchanged and retained. Whilst the computational power available to an artificial intelligent system is great and growing ever larger, an approach of attempting to retain everything and process everything seems likely to fail, in the same way that the scenario of Section 6.3.1 suggests if we attempt to communicate everything we will saturate networks and the recipients processing time. A badly managed belief base was found to cause problems in the scenarios of Section 6.1, and we see a need to manage this whilst providing the agents with an appropriate depth and breadth of facts about their environment. We consider that the SA approach may be a way to address this, where a higher level comprehension belief may be able to replace hundreds of lower level perceptions, but this is a subject of further work discussed in Chapter 7, and we acknowledge that whilst we may be raising more questions that we resolve, we are beginning to identify what further questions need to be addressed.

#### 6.4.4 Constructing distributed systems

The work presented in this chapter demonstrates a variety of software components, coupled in such a way so as to retain general application outside of the problem domain of intelligent vehicles, whilst being able to show relevant, meaningful findings within the field of intelligent vehicles.

A common theme amongst all of the scenarios presented in this chapter is that of knowledge and understanding, with communication at appropriate levels (considered in the SA context) investigated as means to reduce the quantity of information exchanged. The perception experiment results of Section 6.3.1 show that just a blind ‘fire and forget’ of publishing all possible information can work, but risks overloading both the infrastructure and the agent, with the preliminary findings in Section 6.1 showing issues that can occur with an overly large agent belief base (as discussed earlier in Section 6.4.3), and plan failure if data rates are too high.

The results of Section 6.3.1 show that for the perception scenario, switching from a data-push to a data-pull could reduce agent messages by 90% (whilst still achieving satisfactory domain performance), followed by Section 6.3.2 where the results based on communicating at a different level of knowledge suggest that the convoy performance and remain the same (if not better) whilst communication is reduced.

A strength of the distributed system approach is shown in a number of these scenarios, where the Area Of Interest (AOI) module is used to process a large amount of data, generate higher level SA meaning based on this, and publish back to the framework for other components (i.e. the intelligent agents and institutions) to consume. Whilst the implementation of this capability is fairly basic in the current solution, another strength of the framework is that this can be easily replaced with a more advanced implementation (e.g. stream reasoning or improved data fusion). This approach allows members of the framework to process data at an appropriate level (e.g. to allow the intelligent layer more time for reasoning, than data processing) where computationally intensive and non time critical activities can be performed by other components. This is explored further in the accident management scenario presented in Section 6.3.6, where multiple institutions are shown interacting to resolve different aspects of the issue (i.e. deciding if a crime has been committed, handling any insurance claims, as well as dispatching police and managing the vehicle merge). We see this as a benefit of the distributed framework, as additional interfaces can be introduced to expand the breadth of the solution (i.e. extra sensors, processing capability, knowledge stores) for existing components to draw on.

Finally, we revisit the topic introduced earlier in Section 6.4.1, of the need to publish different data types at different rates. Whilst earlier results shown in Figure 4-18 showed message exchange reliably at 1000 messages a second, this was achieved in quite idealised conditions, with only one subscriber, with a server running on dedicated hardware, all communicating over a local area network. Considering application to other domains, it was shown earlier in Figure 4-7 that a single quadcopter could transmit 300 messages a second, and furthermore alternative domains may have a requirement for different message payloads (e.g. audio, im-

agery). Additional research projects based on the BSF framework have demonstrated its use for transferring images encoded as RDF messages, and drawing on the work discussed earlier in Section 4.1.1 along with the results discussed regarding message volume, we summarise that the framework has been shown to support a variety of hardware types, software languages, interface types (e.g. Institutions, the Area of Interest module, simulations), message types and message volumes. We see a significant potential application domain based on this capability in the area of smart homes, which was introduced in the discussion of the Area of Interest module in Section 4.1.5, where different device types, software components, and message exchange may be required. Furthermore, the flexibility provided by the adopted publish-subscribe approach in decoupling components in terms of location, time, and awareness of the existence of each other assists in adding support for further potential domain uses. Currently in the early stages of research are additional BSF components such as facial recognition, motion tracking, and VoIP telephone interaction, with an intention for such components to sense and publish further details about an environment for other BSF members to process.

In general terms, we see a requirement to seek efficient communication approaches were possible (e.g. adopting an SA approach and exchanging less frequent, higher level messages) whilst acknowledging that alternative domains and component types may introduce different communication requirements. Therefore, to address the problem statement of Section 3.2 concerned with retaining a generalised solution, the framework supports a range of message content types, interfaces, and hardware/software types (as discussed in Section 4.1.1), as well as allowing components to communicate at different frequencies on different streams.

## 6.5 Summary

In this chapter, the results of experimentation aimed at investigating the problem statements of Section 3.2 are presented. Initial issues and their solution were discussed in Section 6.1, followed by results based on the scenarios put forward in Chapter 5.

As the approach to the generalised problem topics is based on experimentation in the domain of intelligent vehicles, it follows that the core set of results are focussed on this area. This in itself demonstrates that the framework of distributed components is able to perform meaningful experimentation in a complicated problem domain, where scenarios based on real-life situations have been implemented with results seeming both plausible and interesting. Further to the domain findings, we believe that the implications of these results can be abstracted back to the general problem statements. The situational awareness approach to knowledge representation has been explored directly in two scenarios, where domain performance in a convoy scenario improved when higher level information was exchanged. This was taken further, to exchange information based on the projection element of SA, again with improved domain performance. Whilst we do not expect this to hold true across all applications and areas, it does suggest it is possible to communicate less, by adopting higher level SA constructs, and understand more about the environment.

Following this, a set of more complex domain scenarios were explored, drawing on the use

of institutions to resolve ambiguous behaviour where the socio-cognitive agents struggled to understand the meaning of another human's behaviour. The use of institutions is explored further as a means to target obligations to specific intelligent agents, which may be contrary to their immediate desire, but improve the overall group behaviour.

Having presented domain specific, quantified benefits based on this framework, and drawn out how these relate to the more generalised problem areas, we now move on to review areas of further work (in Chapter 7) which have been identified throughout the course of this work, before offering conclusions (in Chapter 8) in which we review our achievements against each problem statement and objective.



## Chapter 7

# Further Work

In Chapter 6 the results of scenarios developed to investigate the problem statements of Section 3.2 were presented and analysed, both in terms of findings related to the domain of intelligent vehicles and in generalised terms of the broader problem areas of interest.

During this analysis phase, areas have been identified where the work could be developed further, in order to add a deeper understanding of problem topics and to broaden into wider problem discussions. These areas are presented in this chapter, where we consider them as a number of grouped areas.

In Section 7.1 further work directly related to the scenarios is discussed, where we consider improving the quality of the simulation and data extraction, adding further variety to the scenarios, validating the behaviours currently used to represent human drivers, and address specific unanswered questions which arose during the analysis of the scenarios in Section 6.3. In Section 7.2 we consider further work which would explore the problem statements of Section 3.2 by adding human drivers into the simulation, and the possibility of extending the scenarios to include real vehicles. Finally, in Section 7.3 we consider additional problem questions which this work has identified: what factors affect trust in artificial socio-cognitive systems, how to manage dynamic information exchange, how to handle institution directives being ignored or violated, and what time considerations are required regarding intelligent agents and institutions.

### 7.1 Improving intelligent vehicle scenarios

In the problem statements of Section 3.2 we set out a number of areas for investigation, including the problem of how to demonstrate a solution in a specific problem domain whilst retaining a generalised approach. In tackling this, a trade off has to be made between too much focus on the specific vehicle domain against demonstrating meaningful (in the sense of valid, plausible) results. In Section 4.1.3 alternative vehicle representations were discussed, shifting from a highly idealised, simple vehicle representation to more sophisticated vehicle and traffic simulation software. Whilst the more simple representation serves a purpose, the more advanced simulation offers interactions (e.g. traffic lights, vehicle dynamics) and data sources (e.g. fuel consumption,

vehicle emission) which add confidence in the results of the scenarios, and conclusions based on these. This raises the question of whether further improvements to the representation of the vehicle and traffic domain could add value, though even if not, we discuss them here to show our awareness of these possibilities and their impact on the work.

### 7.1.1 Quality of simulated and extracted data

In Section 6.2 a range of metrics developed to capture performance data were presented, which were used during the analysis of the scenarios presented in Chapter 6. Whilst some of these retain general application outside of the problem domain of vehicles, as they focus on the network layer (i.e. message rates, delivery delays), the majority have had to be developed specifically to capture vehicle and traffic performance characteristics.

Whilst most of these metrics proved sufficient to identify both individual and overall group performance, in the variable speed limit scenario in Section 6.3.5 a more sophisticated metric could have helped identify congestion indicators more clearly. Similarly for the post-accident management scenario in Section 6.3.6, more sophisticated metrics may help identify the overall impact of the accident and therefore the benefits and shortcomings of the proposed institution approach. Here we encounter a problem in the challenge of knowing what to measure before the simulation is performed, which is where the logging and SPARQL analysis detailed in Section 4.1.6 was planned to offer assistance. However, this links to the discussions regarding the problem of what, and how much, to communicate, as the logger utility can only store messages which have been published to the framework. This is discussed in more depth in the following Section 7.3, in the form of intelligent agents needing other components to broadcast more or less data, and of different types, which is similar to post-simulation metrics requiring information that may not have been generated. For example, following a simulation run, it may be proposed that the use of brake lights could be used as an indicator of congestion, but if this data was not extracted and published then it would not be present in the logged data, and so would require the simulation to be re-run with that data published.

### 7.1.2 Variety of scenarios

The scenarios described in Chapter 5 represent a number of traffic and vehicle contexts which are put forward to investigate the problem statements of Section 3.2, which in themselves are chosen to represent problems faced by artificial socio-cognitive systems. In effect, this follows a top-down approach, where we have worked down from the high level general problems, into a specific domain, and identified issues there (based on existing issues in vehicles and transportation). Having established in the results of these scenarios in Chapter 6 and found that the use of intelligent agents coupled with an institutional framework can assist in such problems, it would be interesting to switch to a bottom-up approach, selecting a wider range of issues from the problem domain and addressing them with the framework that we have established.

This needs to be weighed against the risk of becoming too domain specific, an effort on which we have attempted to retain focus throughout this work. However, the counter argument would

be that having integrated both traffic simulation and a number of metrics for that domain, the additional effort in developing new scenarios may not be significant, and would provide further insight into issues in the generalised problem area. Identifying situations where the proposed approach fails would provide a useful insight into where the agents, institutions, or framework itself has some shortcomings for a particular type of scenario.

### 7.1.3 Validity of behaviours

Of the further work considerations which focus on the problem domain of intelligent vehicles, an area in need of validation and re-assessment is that of driver behaviour, i.e. the representation of humans in the simulation. In the variable speed limit scenario discussed in Section 6.3.5 and the accident management scenario in Section 6.3.6, the effectiveness of the agents and institution approach is gauged largely on traffic congestion measurements, but in its current form the traffic simulator does not model crashes or congestion behaviour based on poor driving technique.

The approach taken to mitigate this lack of representation was to modify vehicle (e.g. maximum brake and acceleration rates) and driver (e.g. impatience, reaction time) parameters to the point where expected congestion behaviour could be seen. This introduces the risk that vehicles may be showing surface-validity without any valid underlying models (from a human psychology perspective), and even in this case, the surface validity is only reviewed by a non subject matter expert, rather than those involved in the specific domain of vehicle congestion patterns. The impact of this is that if the modelling or parameters are based on invalid assumptions, then conclusions drawn on the results of the institution in affecting that scenario may also be invalid.

However, introducing credible human modelling is a challenging topic (e.g. introducing human behaviour into simulations as considered in Spear and Baines [2009]), requiring a significant amount of time and human factors expertise. It may prove a better use of effort to present the scenarios to an appropriate expert audience, or to seek captured data of traffic congestion from other sources, than to develop such specialised models in the simulation.

### 7.1.4 Remaining questions

In the scenario analysis put forward in Chapter 6, there are some results which prove difficult to explain given the current experimental configuration and set of developed metrics. We highlight two of these cases, where the findings suggest more than just a refinement or improvement is required, instead posing more general questions regarding the adopted approach.

The first of these arises from the waypoint scenario detailed in Section 6.3.2, where the message and percept counts were higher than expected, and very similar to the earlier data-pull approach in Section 6.3.1. Whilst this approach results in less coordination information having to be exchanged (i.e. exchange a list of waypoints at the start of the convoy move, rather than moving to the position of the vehicle ahead each simulation step), it is reliant on all plans adapting to this new approach compared to the previous data-pull. The difficulty is that plan

re-use is made use of in the agents as far as possible; for example, rather than coding a specific `moveTo` plan for each scenario variation, the same plan should cover multiple uses. This can cause problems, as the plan may be using an approach more suitable to a different scenario structure (e.g. asking other agents for information rather than deriving that data itself). This problem is further compounded by the difficulty in the current architecture of identifying what may be triggering such an exchange of data, though the utilities suite developed as part of the framework could be extended to assist in such analysis. Such development may help in debugging the system (and relates to the problem statement of observability of data), extending the question further to be not just what is being communicated, but who is communicating and how much. The ability to identify such data then lends itself to the issues discussed in Section 6.4.1 regarding the ability for one component to ask another to publish more, less, or different data streams, where the desire of one component may be to the detriment of others (e.g. requesting a key publisher sends data at ever increasing data rates), and improvement in the observability of this part of the framework may assist in supporting a communication management institution (as discussed later in Section 7.3.1).

Another unanswered question remains in the variable speed limit scenario presented in Section 6.3.5, where in the variation of the scenario with an institution involved in slowing vehicle speeds, there was undesirable behaviour when the institution obligation expired and vehicles attempted to return to their previous behaviour. There are aspects of this which relate to the temporal nature of institution involvement, and the need to manage an agents return to their previous behaviour, which is discussed later in Section 7.3.3, but there are also aspects of this relevant to the scenario itself. Earlier in Section 7.1.3 the benefit of validating driver behaviours in the simulation was discussed, and doing so would identify whether the unexpected behaviour is due to the driver, or how the institution has managed the situation. For the latter case, it may be that tuning the institution parameters (in terms of obligation duration, how many vehicles to apply it to, and what speeds should be set), could lead to a more successful (in terms of domain performance) set of results for this scenario. With the open framework of the BSF, it would be possible to develop an additional module to provide a generic capability, e.g. a genetic algorithm based approach to tune these parameters over multiple simulation runs and demonstrate improved domain results.

## 7.2 Exploring further applications

In the previous Section 7.1 we presented areas where the scenarios could be extended or improved to explore further or confirm existing findings specific to the problem statements of Section 3.2. The domain of intelligent vehicles was selected as an appropriate area in which to ground research into these problem statements, offering sufficient complexity and real life issues which could be addressed by the proposed framework.

Some further applications for this work have already been introduced, where in Section 4.1.3 alternative target devices of real vehicles and quadcopters were explored, along with initial work in integrating them into the BSF framework. In addition to these ideas, we also propose

the introduction of human controlled vehicles into the simulation as a means to explore further applications related to the same vehicle domain. Details of these additions, and the motivation for undertaking such extensions, is now presented.

### 7.2.1 Adding human in the loop capability

In this topic of further work, the additional functionality would inject a vehicle into the SUMO simulation in a similar way to how agents insert their vehicles, but rather than an intelligent agent controlling that vehicle, it would be controlled by a human driver. As a 3D module has already been developed, this could be re-used to provide the in-vehicle view to the human driver, and handle their control signals intended for the vehicle. There would be challenges in transforming these into appropriate messages for the SUMO simulator, but a solution could be found.

This would enable further investigation of the problem statements, with particular benefit to the issue regarding observability. Firstly, the intelligent agents would have to deal with more challenging ‘imperfect’ humans to reason about (such as incorrect or missing signalling of intent, ambiguous communication, aggressive or defensive driving techniques), which may then give rise to further findings against the problem statement of resolving problematic situations. Additionally, the human control signals over the vehicle could be made observable across the BSF framework, which would raise two further possibilities: displaying these signals in the 3D view to aid understanding of scenario events, and to investigate whether supplying these signals to the intelligent agents improves their reasoning of what the human is about to do. Considering the introduction of intelligent vehicles to the roads as a form of novel technology (as presented in the problem statements of Section 3.2), it would be interesting to see whether retro-fitting human driven vehicles with some device that relays their human control signals over vehicle to vehicle communications, to the intelligent vehicles, may improve the agents’ understanding of the environment.

### 7.2.2 Extending to alternative devices

The concept of interfacing with real vehicles and quadcopters was introduced earlier in Section 4.1.3 as a means to assess how de-coupled the intelligent layer is from the entities it controls, and to test the framework with live data rather than only simulated entities.

The motivation to revisit this as further work is to assess how well the developed framework can be transitioned to another application, and to assess if our findings hold true in a similar domain situation, in a similar argument to that of Section 4.1.3 for the representation of motorway and city contexts in order to assess performance against a wider variation of speeds, junctions and dealing with multiple lanes.

The use of real platforms (i.e. vehicles and quadcopters) would move away from idealised (in terms of physical response, reliability of communication) representation, and when a similar activity was performed in replacing the simple `xmppVehicle` with the SUMO simulator (both described in Section 4.1.3) a useful set of challenges and findings emerged. We would expect

the inclusion of real devices to introduce additional complexity to the scenarios, and provide a more strenuous test of the intelligent layer and institution structures when managing real platforms.

Development based on additional device types such as the Android, Raspberry Pi, and Intel Galileo boards is also underway (as discussed earlier in Section 4.1.1), with the intention to add further components in order to demonstrate the use of the BSF framework in a wider range of applications, with initial focus on the smart home domain.

Such work provides insight into the problem statement regarding distributed systems providing a variety of components for generalised applications, both in how easily replacement components can be integrated and how transparent this is to other existing components (i.e. how well has the system remained uncoupled, or have specific inter-component dependencies arisen).

## 7.3 Further problem questions

Whilst a number of problems and issues faced by artificial socio-cognitive systems were put forward in Chapter 3, this is not considered an exhaustive list, and even where these have been specifically investigated there remains breadth within each problem statement which has been unexplored. In this section, we present additional questions which have arisen in the course of this work, the investigation of which would add to the understanding of issues faced by artificial systems and areas where the proposed framework may be of benefit.

### 7.3.1 How to manage dynamic information exchange

In the percept-based scenario presented in Section 6.3.1, the question of how much data should be published to improve visibility of internal states arose. There was a contrast between two sets of results with one (Figure 6-6) showing up to 6000 messages exchanged within the internal agent framework, and the other (Figure 6-7) showing significantly less published external to the BSF framework. This leads to the question, of how much data *needs* to be published in order to improve understanding, with links to the SA concepts as a means to distinguish between high level information where one message may provide a richer source of understanding than a number of low level messages. However, there is a question of where the decision of message suitability should reside, with the earlier premise discussed in Section 3.4.1 that the publisher can send to different data streams, and subscribers then select whichever data stream they require.

Further work is proposed to develop a dynamic ‘tuning’ of message exchange, allowing a subscriber to request that a publisher sends at a higher or lower frequency, or to request it starts publishing messages of a different type (e.g. start publishing current fuel efficiency). This has some overlap with the SA concept, as from a global collective perspective, it may work out more efficient for an entity to publish some higher level SA fact (which another entity desires) than for entities to attempt to reason about lower level received percepts. For example, in the vehicle

domain, vehicles could attempt to infer the destination of a vehicle given its speed, previous positions, and so on, however it would consume less computational resource overall for this vehicle to simply publish that piece of information. However, if entities demand further and further details from one entity, it may become detrimental to that entity to dedicate significant (self) computational resource to provide that data. This is an area where we consider a need for a communication institution, to regulate such demands and ensure entities are not providing too much (and consuming resources as well as saturating networks) or too little (saving self resource but impacting other entities) information, where such work has analogies to the ‘freeloading’ issues presented in Balke et al. [2011] (as discussed in Section 2.2.2). Such an institution may also be able to assist agents beyond regulating message quantities, for example handling an agent query of how to get information about a particular subject. We propose the construction of such an institution, along with required functionality within the framework, in order to identify how this capability could function, as well as assess any benefits it might introduce.

### 7.3.2 When directives are ignored or violated

A number of the scenarios presented in Chapter 6 feature institutions which issue obligations for intelligent agents to implement. Inherent in the institution model is the functionality to handle violations when obligations are not met, but in the scenarios explored in this work the intelligent agents are compliant with the requests: a car moved out of the way after a headlight flash, vehicles slowed down on receiving a new speed limit, and vehicles merge lanes to get past an accident. For all of these cases, there could be reasons such as safety (i.e. would require excessive braking) or obstructions (i.e. a vehicle in the way in the desired lane) which would justify the agent not implementing the obligation. However, there are also cases where the obligation may not be obeyed which would not be considered justified, such as not slowing down in the variable speed limit because it would increase the time to arrive to the destination.

The further work proposed here is to explore the handling of such missed obligations, as specific to the problem statement regarding balancing individual needs versus the greater social welfare, there is a need to demonstrate some penalty or deterrent for ignoring obligations, while handling cases where it is appropriate for the obligation to be ignored. In addition to this, some re-planning needs to be performed (possibly by the institution and the intelligence layer) in order to mitigate the impact of the missed obligation. The construction of additional scenarios could be performed in the existing domain of intelligent vehicles, in order to explore the intelligent agents decision making process of whether to follow an obligation, as well as the institutions’ ability to handle such an event and any associated re-planning.

### 7.3.3 Time considerations

The final further work topic relates to the time dimension of the problem area, where in the course of the experimentation carried out in this work three areas have been identified where temporal factors have impacted the performance of the system.

The first relates to the temporal element of beliefs, as introduced in Section 6.4.3, where

issues had been found in the expiry of beliefs (e.g. the vehicle has moved to a new position, but still held the belief of the previous position as well as the new position). This issue relates to the general consideration of knowledge and understanding in artificial systems; that whilst the expiry of some beliefs makes logical sense, maintaining awareness of old beliefs may help in reasoning (e.g. expiry of `currentPosition` belief, but if these beliefs remain accessible then the agent could reason about where it has travelled from). This has some relation to the earlier discussion in Section 7.3.1 that in general terms it is difficult to know ahead of time what information may be required. We propose additional scenarios are used to investigate various approaches, from the case of agents retaining as much information as possible and so having a large historical belief set to reason with, versus agents with an aggressive purge of beliefs but with risk of reducing reasoning capacity about historical events.

The second area of further work concerns the temporal element of institutions and their obligations, relating to findings from the variable speed limit scenario presented in Section 5.6. In this scenario it was found that obligations may persist for some time period (in the case of this scenario, vehicles maintain a reduced speed for five seconds), and once the obligation has expired there may be a need for the institution to manage the return of the entity to some steady-state. Whilst how an obligation is implemented is a choice made by the intelligent agent, there is the question of whether the agent should also choose how long this obligation lasts for, or whether this should be specified as part of the obligation issued by the institution. Further scenarios could be constructed to explore this issue, as it seems likely that the approach may need to vary in different cases.

Finally, there is the general issue of the permanence of the institution itself. In this work we have explored the approach where institutions are instantiated temporarily in a continuous environment, and dissolved once the situation is resolved. This has the draw back that any previous institution fluents and states are likely to be lost, but this may not necessarily be an issue. However, further scenarios should be constructed here in an effort to identify the merits and appropriate uses of alternative approaches. For example, as junction control via traffic lights would be an ongoing exercise this may be suited to a permanent institution, but for resolution of incidents such as the post-accident scenario, it may be appropriate to organise a temporary coalition of institutions until the issue is solved.

## 7.4 Summary

Firstly, further work relating to the scenarios themselves was put forward, to improve the quality and confidence in the validity of extracted data, to widen the scenario contexts, and to address any outstanding questions left over from the analysis in Chapter 6. Following this, further work in extending the framework to introduce human controlled vehicles, as well as real vehicles and alternative air platforms was identified, in order to assess how generalised the approach has remained, and to identify if the framework can assist in introducing novel technology to the current human driver environment.

Finally, a number of further problem areas were put forward, regarding the management



of information exchange, how to handle institution guidance not being followed, and the time dimension such guidance should include. We believe addressing such additional questions would improve the understanding of how the approach of intelligent agents and institutional frameworks could work in practise for a generalised range of applications. We also reflect that a number of areas of further work have been put forward, not because of a lack of content of the research presented in this work, but because the problems faced by socio-cognitive systems touch upon a broad range of issues. These issues are likely to all need addressing before such systems could be used in live situations, but this is reflected within the problem domain of autonomous vehicles (seemingly in a similar position), where legal questions around how such vehicles would operate, public confidence in these systems, and required infrastructure capability, all need consideration.

The final chapter of this work is now presented, where conclusions are drawn from the findings of this work against the original problem statements, objectives, and challenges put forward.

## Chapter 8

# Conclusions

In this work, we have been considering the challenges faced in the construction and deployment of artificial socio-cognitive systems, discussing general problem areas in Section 3.1 of: i) knowledge and understanding, ii) human-machine interaction, iii) the introduction of novel technology, iv) the ability of intelligent systems, and v) constructing distributed systems. We then established specific problem statements in Section 3.2, before grounding these problems in the domain of intelligent vehicles, discussed in Section 3.3.

In Chapter 5 we presented a number of scenarios to investigate these areas, firstly exploring different levels of situational awareness, with vehicles exchanging low level perception position information compared to higher level route information, and the impact of introducing higher level projection on reacting to the future state of traffic lights. We then explore scenarios where to assist intelligent agents in their tasks, an institution is used, first dealing with a socially ambiguous event for which the agent has no plan, then in speed management where the collective benefit requires action contrary to the individual desire, and finally where the coordination of multiple bodies and entities is required as part of a post-accident management scenario.

The analysis of these scenarios is presented in Chapter 6, first dealing with the domain specific results regarding the performance of the vehicles in order to identify strengths and weaknesses of the variation between scenarios, and also to demonstrate the achievement of the framework components in generating credible results in the chosen problem domain. We then consider the more general implications and findings based on these results in Section 6.4, and we now conclude our assessment by reviewing our findings against the specific problem statements of Section 3.2.

### 8.1 Findings against problems

A number of problem statements and corresponding challenges were put forward in Section 3.2, which we now revisit in light of the findings from the analysis of the scenarios presented in Chapter 6, drawing on the general findings discussed in Section 6.4 as well as the scenario-

specific findings, to reflect and identify where findings have been made.

### **How to build awareness and understanding of a situation**

The concept of situational awareness has informed a number of design decisions in the construction of the BSF framework and generation of scenarios presented in Chapter 5. We consider information exchange at different levels of complexity and frequency of exchange, and in the scenarios considered, there is evidence to support the intuition that different layers of the solution are able to operate at their own ‘natural’ level (i.e. the intelligence layer drawing on higher level low frequency data, compared to the Area of Interest module consuming higher frequency but low level information).

Variation in communication approaches has been demonstrated in a number of scenarios, where convoys have been used to explore alternative communication strategies considering different levels of SA, shifting from high frequency exchange of low level SA information (in the scenario of Section 5.2, x,y coordinates of the vehicle ahead), to low frequency exchange of more complex high level SA information (in the scenario of Section 5.3, convoy route information). Intrinsic in the exchange of more sophisticated information is the ability to perform more reasoning and improve understanding, for example knowing a vehicles entire route allows calculations around potential journey times, fuel requirements and so on. Such capability is explored using the situational awareness element of projection in the scenario presented in Section 5.4. This scenario explores reasoning around future states of traffic lights, where through a combination of the Area Of Interest module and institutional governance, intelligent vehicles were shown to adapt their speed in order to arrive at traffic lights when they are green, thus reducing fuel consumption. In this sense, the awareness of the intelligent vehicles is extended to include a temporal element, creating an understanding of what the future state of their environment will have on their state.

The exchange of different SA levels of information has to be supported by an appropriate intelligence layer, and in this work we have shown the integration of the Jason BDI framework to provide this functionality, with agents situated in the environment via a sensory and action mechanism provided by the BSF framework. This provides both a generic capability which can be applied to other problem types and domains, and an intelligence layer which is able to make use of received higher level information as investigated in the alternative SA approaches explored in the scenarios of Sections 5.2, 5.3, and 5.4.

Considering the challenges set out in Section 3.2 regarding this problem statement, we put forward an SA based approach as a means to meet the awareness and understanding requirements, where we attempt to exchange knowledge between components at a suitable level for that layer, i.e. with the intelligence layer receiving higher level information rather than being saturated with low level percept information. This approach supports the rest of the problem statements, where by including similar consideration across all components, for example supplying institutions with appropriate levels of information such that they can focus on their task, rather than data transformation activity. We retain application to other

problem domains by developing generalised capability, customised for a specific task through configuration rather than coding, based on a generic core framework.

Finally, we reflect on the corresponding objective from Section 1.2, which was to demonstrate appropriate intelligence. We consider this in terms of our performance in the domain of intelligent vehicles: were the agent controlled vehicles able to achieve their task as well, if not better, than the existing normal vehicles? Whilst it would be easy to consider the scenario results shown in Chapter 6 as a successful demonstration of appropriate intelligence, some caveats are required. This work has been based on simulation, and whilst there has been effort to base this on a known simulation package with reasonable fidelity of modelling, this is a different case to operating in the real world amongst humans. However, we are interested in demonstrating a *capability* rather than a final *solution*, as we are not domain experts in the area of intelligent vehicles, and we also wish to demonstrate an approach which retains a general application.

### **How can the reasoning process and information flow behind a decision be made observable**

This problem statement focusses on how we can make available ‘inner’ working of an agents deliberative process to other components within the BSF framework, to improve their capability of interacting with the intelligent agents (e.g. the institution framework) as well as to support the system designer in debugging during the design and test phase (e.g. presenting agent state data along with geospatial states in the 3D viewer).

The approach to address this has been built on two key areas: observability of message exchange, and observability of component internal state data. To address the observability of message exchange requirement, the approach set out in Section 4.1.1 has been to build on an existing Bath Sensor Framework (BSF) which provides a distributed messaging framework, based on the exchange of semantically annotated data. This allows BSF components access to exchanged data and assuming that a common ontology is used for the semantic annotation, supports components in understanding and analysing this data. However, this assumption could be explored in further work, shifting from a homogeneous configuration of BSF components to a more challenging arrangement in order to investigate how heterogeneity can be supported. This raises a requirement of being able to recognise this non heterogeneous state, for example whilst RDFs may support the definition of base units (e.g. speed, distance), there is the question of how less specific terms (e.g. bright, slow) could be handled. Furthermore, there is the requirement to then manage this variation, for example it is possible to envisage multiple institutions coming into conflict where heterogeneity occurs, but still needing to interact and provide a resolution.

For the requirement regarding the observability of internal component states, the approach adopted was to modify key components such that they publish additional information to the BSF which would not normally be accessible. For the intelligence layer, this involves publishing information regarding the mind-state of individual agents regarding their beliefs, received messages, and plans. For other components, this involves extracting similar internal data e.g. the

traffic simulator publishing simulation time and vehicle route information. The difficulty becomes, how do we (as the system designer) know ahead of time what should be published? Even knowing this, how frequently should the information be sent? More detailed examination of this follows in the next problem statement regarding finding the balance between quantity and appropriateness of data, and so here we focus on the *how*, and whether we have demonstrated a mechanism which improves the observability of the system.

Considering progress in tackling the problem statement, we revisit the associated challenges. Firstly, the challenge of identifying a suitable approach to improve the observability of message and internal state data. In Section 6.1 the immediate benefit of the availability of such information was shown, where the 3D view component is able to represent internal agent beliefs and reasoning (e.g. an agent belief of its destination, what it has calculated to be its collision zone, and its last received message). Secondly, that the messages can be handled by both system components and humans debugging the system. The adopted approach of using semantically annotated messages has been shown as effective in the scenarios reviewed in Chapter 6, and in supporting debugging activity in Section 4.2 and Section 5.1. However, this is dependent on the human debugger having an awareness of how the system operates, as well as understanding the ontology used in both the messages themselves and any annotation around them.

The final challenge concerned operating at a variety of transfer rates, as the adopted approach to improve observability also needs to work within infrastructure limitations in order to have real-world application. Whilst consideration is needed for the quantity and appropriateness of message exchange (addressed in the following problem statement), we show the framework operating with over 750 messages a second in Section 4.2, and operating with over 300 vehicle entities in Section 6.2. However, we have also shown how a single entity operating in the real domain can introduce 300 messages a second in Section 4.1.3. We believe that the capability of the framework has been shown to be sufficient (and may be improved with further effort e.g. tuning and federating XMPP servers to distribute load), as long as design decisions are taken not to saturate it (e.g. all simulated vehicles publish their position information every 0.1 seconds). This discussion falls into the area of appropriate message exchange which follows in the next problem statement review.

Finally, we revisit the corresponding objective from Section 1.2, to demonstrate improved observability of reasoning and exchanged information, which we believe has been shown throughout the scenario results presented in Section 6, through i) the capability of being able to debug the observed behaviour (e.g. in Section 6.1), ii) system components being able to interact at an appropriate level of complexity and perform their role (e.g. in the scenarios analysed in Section 6.3), and iii) being able to extract domain performance measures through the metrics presented in Section 6.2.

### **Finding the right balance between quantity and appropriateness**

The topic of what to communicate and at what frequency has occurred throughout this work, and we have explored an approach based on communicating at higher levels of SA with specific

scenarios in Sections 5.2, 5.3 and 5.4 to assess vehicle performance when taking alternative communication approaches. Whilst in those specific scenarios, vehicle performance was the same, if not better, this introduces the more general case that any such balance is context specific, both to the domain itself and particular cases within a given domain.

The concept of communication streams is discussed as an approach to allow components which desire a specific type of information (e.g. high volume simple facts) to subscribe to it, and whilst the architecture presented in Section 4.1.1 supports this, in Section 7.3 we discuss the need to vary message rates of streams as well as the type of information published, as consumers of data may have changing requirements as the scenario evolves. At an architecture level, this concept appears again in Section 7.1.1, that in order to log everything for post-simulation analysis, everything needs to be published. The domain specific nature of this problem is highlighted in the quadcopter work discussed Section 4.1.3, that whilst SA exchange of higher level information may help in some cases, some components may need high frequency, low level data, e.g. to track quadcopters navigating around a city.

Whilst we have been unable to put forward a definitive solution for this problem, we highlight a beneficial design finding, and a related piece of further work. We find that by considering information exchange in SA terms, and seeking to exchange higher level information where possible, benefit is shown in the scenarios of Sections 5.2, 5.3 and 5.4. In these scenarios, we find that vehicle behaviour is improved (showing less braking and acceleration) when performing a convoy move if communication is based on higher level information of the intended route, rather than exchanging low level, higher frequency position information in a follow-the-leader approach. We also find that when using high level SA information regarding the future state of the environment, we can reduce communication further and demonstrate a reduction in fuel consumption. These findings address the corresponding objective for this problem statement (put forward in Section 1.2) to explore the effect of alternative communication strategies.

Further interest lies in demonstrating the concept of dynamic communication negotiation as discussed in Section 7.3, since any balance between quantity and appropriateness is likely to vary, even during a scenario, but certainly when considering a wider set of potential applications. As entities and components may not know ahead of time what information they require, and at what frequency, the ability to negotiate this in realtime would be an interesting development to pursue.

### **How to resolve problematic situations**

In the problem statements of Section 3.2, we put forward examples of problematic situations where an intelligent agent is unable to bring about a solution to a problem, lacks knowledge or appropriate plans, and may lack self-awareness that it does not know what to do. To help address this problem, we propose introducing a governance framework as outlined in Section 4.1.4 where issues are referred to in order to bring about some resolution.

Whilst the capability this introduces can be highlighted in a number of ways, we specifically demonstrate this in the scenario put forward in Section 5.5 where an ambiguous action of

another driver (flashing of headlights) occurs, and the intelligent agent has no suitable plan to handle this event. With no institution active, the intelligent driver takes no action, but with an institution present the intelligent driver receives an obligation to move out of the way. Within the domain of intelligent vehicles, we see this as a general mechanism to resolve issues which may be contextually or culturally specific, where it would be challenging to equip the agent with the capability to handle all permutations of events, and instead draw on the institution as a late-binding mechanism to assist agents in such cases.

Two associated challenges were put forward in Section 3.2 for this problem statement, where the first concerns the need to define domain specifics in a general fashion. This is addressed inherently in the selected InstAL institution representation, with logic programming providing an abstraction away from domain specifics, and whilst domain terms and outcomes are specified in configuration files, the institution component itself retains a general application. The second challenge concerns the integration of this capability into the wider framework, which requires both the coding effort to enable information flow and instantiation, but also appropriate information exchange, where we refer back to the SA concepts. In the results of Section 6.3.4 we show the use of an institution to provide a late-binding mechanism for the intelligent agents, demonstrated in a case where the agent is unaware of a social convention (to move out of the way when another vehicle flashes their lights). In this case, the agent is unaware it needs to take any action, and the assistance offered by the institution allows us to address the objective of Section 1.2: to provide assistance to an agent where it has no suitable plan.

### **How to balance individual pursuits versus the greater social welfare**

This problem statement is concerned with the issue of autonomous intelligent agents pursuing their own goals to the detriment of the overall collective, and how such behaviour can be managed. There is the related possibility that the coordination of a number of individuals who may not have acted (or even known to act) together under their own choice, may result in an improvement to the overall society.

We explore a number of scenarios based around such issues; in the scenario of Section 5.4 a vehicle is told to slow down on approaching a traffic light in order to reduce fuel consumption (as an environmental benefit for society), in Section 5.6 a number of vehicles have their speed reduced in order to ease congestion (to improve traffic flow for the rest of vehicles) and in Section 5.7 vehicles are coordinated in merge lane behaviour in order to navigate past an obstruction. These scenarios address the corresponding objective for this problem statement set out in Section 1.2, regarding changing an agent's pursuit of its own interests for the greater good.

The resolution of these scenarios is managed by the institutional framework described in Section 4.1.4, where we draw on the institution's capability to issue obligations to specific individual vehicles, in order to bring about the desired result (i.e. to benefit the collective). Whilst the scenario results presented in Chapter 6 shows a positive effect of this intervention and serves to demonstrate the approach, some significant questions remain and are highlighted

in the further work discussions. The first of these is discussed in Section 7.3.2, regarding the handling of ignored obligations. It may be perfectly valid in some cases for entities to not comply with institution obligations, and is a strength of the approach in providing a handling mechanism for this eventuality, the handling of such cases needs to be explored. Whilst penalties may serve as an appropriate consequence if the obligation was willingly (and inappropriately) ignored, cases where it was correct to ignore the obligation (e.g. due to safety reasons) would provide useful insight, and establish what recovery and retry mechanisms could look like.

The second question requiring further consideration is introduced in Section 7.3.3 regarding the temporal nature of obligations and how this should be handled. In the variable speed limit scenario results presented in Section 6.3.5, it was found that there could be improvement in both the duration of the obligation, and the return to the entities prior state. This raises questions of where the responsibility lies to assess if an obligation has been satisfied and should be expired, and whether having diverted an entity from a previous state, an institution should manage the entities' return to that state, or allow the entity to control this process.

Considering the challenges outlined for this problem statement in Section 3.2, the scenarios related to this problem demonstrate guidance being provided to artificial entities resulting in an improved result for the wider collective, and whilst the autonomy regarding a specific feature of that entity (e.g. speed selection) has been affected, the overall autonomy remains (and indeed, the autonomy to disregard the obligation). The final challenge highlighted of returning an entity to its prior state is a subject where we consider some questions remained unanswered, regarding: i) the duration of obligations, ii) where responsibility lies in determining whether an obligation has expired, and iii) how an entity should return to its previous condition after an obligation, which we suggest is investigated in further work through the development of additional scenarios to explore contrasting approaches.

### **How to demonstrate the application of the principles we have identified, whilst retaining a general solution**

This aim has been considered in all the scenarios put forward in this work, where the desire to demonstrate a measurable impact in the problem domain has been offset against developing heavily bespoke approaches specific to that domain. At a high level, the BSF framework has been shown to bring generic capabilities together through the course of this experimentation, and that in itself offers a general solution, as there is no heavy coupling (e.g. API calls or specific communication approaches) between components, instead a publish-subscribe of RDF messages where if one component is substituted (e.g. traffic simulator replaced with some other simulation), the rest of the BDF components would still receive published data.

This form of abstraction permeates each component, allowing software modules to best use their own strengths in adding capability to the overall BSF aims. For example, the institution remains a generic capability, configured for certain scenarios through the use of definition files, which can be replaced or customised for other domains or applications. The functionality offered by the institution remains the same: the ability to provide an external governance structure



and issue guidances which BSF members are able to consume and act upon. Likewise for the 3D component presented in Section 4.1.7, whilst this is currently configured for two locations in the UK and uses vehicle specific models, the generic capability of displaying agent mind state data, institution interaction, along with additional data overlays, remains generic regardless of the chosen problem domain.

The social interaction retains a general application outside of the intelligent vehicles problem domain. In this work, vehicle behaviour is assisted by the institution offering social guidance and resolution of situations where the agent has no relevant plan for what to do. Whilst institutions have been demonstrated as having a positive impact in the scenarios presented, the challenge of how to handle both agents and humans not complying with such obligations has not been addressed. This was discussed in the further work of Section 7.3.2 with the proposal to construct additional scenarios and explore how obligation violations can be handled in the domain of intelligent vehicles. This raises the question of how humans would be included in such an approach, though we consider this an open problem in artificial socio cognitive systems development. For the problem domain of vehicles, whilst the obligation could be communicated to the human driver (assuming approach network and display equipment), thought would be needed whether this could be a feasible approach (given privacy concerns, the likelihood of rejecting a ‘big brother’ governance structure, and so on) or whether such systems should focus on managing intelligent agents only, with the capability to work-around humans in the environment.

### **How distributed systems can support a variety of component types for a generalised set of applications**

In this problem statement we investigate issues associated with adopting a distributed system approach for the framework used in the scenarios presented in Chapter 5. Issues requiring consideration are how components running across multiple PCs over some network infrastructure can communicate and exchange messages of both a suitable type and rate for the proposed scenarios, and how this communication can be assessed by a human observer to support performance analysis and debugging.

We adopt the Bath Sensor Framework (BSF) as the framework used in this research, and have extended to provide additional tools focussed on logging, replaying, analysing and monitoring (presented in Section 4.1.6) as well as a generic 3D capability (presented in Section 4.1.7). These tools provide a generic capability to assess low level network metrics such as message delivery delay and message quantity per second, for realtime analysis in the form of live graphs, or for offline analysis in the form of CSV files and SPARQL database entries. These utilities have been used extensively in the analysis presented in Chapter 6, providing the debugging capability shown in Section 6.1 and the generation of results used in Section 6.3. This set of applications, combined with the components described in Section 4.1, addresses the objective set out in Section 1.2 for this problem statement, to develop a distributed systems solution applicable to a general set of problems. The developed framework and components provide

benefit to multi agent system builders, bringing together a number of tools which have been shown in action throughout the scenarios of Chapter 6. However, an empirical demonstration of the benefit to programmers and debuggers should be considered in further work, in order to better understand and qualify benefits to a wider community.

Increasing sophistication of scenarios was necessary to provide insight into the full set of problem statements, which pulled us into increasingly detailed analysis specific to the problem domain, but helped in demonstrating a wider set of metrics (discussed in Section 6.2) with low development time required due to the generalised approach adopted, e.g. the graphing capability can be quickly modified to display different data in a variety of plot types.

The framework has also been shown to be generic in terms of message content, with components able to extract data of the type they require, e.g. the Area Of Interest module presented in Section 4.1.5 consuming high volume, low SA level information, and publishing higher level information for the intelligence layer and institution to consume. Alternative message formats have also been introduced, where whilst the RDF format is used in this work, the JSON format provides an another option as discussed in Section 4.1.1. Exploring this, in conjunction with alternative message payload types (e.g. audio, video) could be a beneficial piece of further work in order to broaden the generalised application use, especially in conjunction with the concept of dynamic content provision discussed in the further work of Section 7.3.1.

## 8.2 Closing remarks

We envisage socio-cognitive systems becoming increasingly prevalent in society, where such novel technology will be expected to integrate seamlessly into a challenging environment, be required to interact with humans and deal with unexpected complex situations, whilst achieving tasks and goals. Whilst this is a challenging objective, approaches available within the computer science discipline offer methods with which to address the problem, and not only solve it, but to add benefit and capability not present in the current solution to the problem. We explore how the individual pursuits of such intelligent agents can be managed to improve the greater collective welfare, providing a guidance mechanism for occasions when intelligent agents lack appropriate or awareness knowledge of what to do.

It has been the aim of this work to retain a generalised approach throughout, whilst demonstrating application and benefits within the domain of intelligent vehicles. Having developed a framework based on observable, semantically annotated information exchange, where components can be replaced transparently by other components, and with some initial exploration of alternative real world land and air based vehicles, we feel the generalised nature of the work has been shown.

We finish with thoughts on an interesting future, based on developments already moving at pace, where intelligent autonomous vehicles are able to drive passengers to their destination with no human control needed over the vehicle. As focus shifts from achieving this for a single vehicle, consideration will likely move to their collective behaviour, where we see the role of governance structures introducing some significant changes. Traffic flow becomes an externally

managed coordination problem, where vehicle speeds are adapted before arriving at junctions, and traffic lights become redundant. Motorway merging and speed management is handled and agreed between vehicles, potentially reducing congestion and improving throughput on roads, thus making better use of, and reducing load on, (expensive) physical infrastructure.

In general terms, the capability to support such work should prove useful as we see an increasing number of devices including some form of communication (WiFi, Bluetooth, Zigbee) capability. Whilst the work discussed in Chapter 7 highlights a number of problems and areas needing further further work, we are hopeful that the approach demonstrated in this research will have a benefit in the emergence, uptake and acceptance of artificial intelligent systems.

# Appendix A

## Additional institution definitions

In this chapter, we present the institution definitions used in the accident management scenario presented in Section 5.7.

### A.1 Crash group institution

Listing A.1: Crash group institution definition

```
% dynamic institution established containing affected parties of the crash
institution crashgroup;

% types
type Agent;
type Victim;
type InsuranceState;
type CrimeType;
type PoliceUnit;
type CollisionLaneMember;
type FreeLaneMember;

% events
exogenous event crashed(Agent, Victim);
inst event intClaimCompensation(Victim, Agent);
inst event intRequestMerge(CollisionLaneMember, FreeLaneMember);

% fluents
%% add agent to mergepairing, as there could be multiple accidents, plus this
    helps resolve info
fluent mergePairing(CollisionLaneMember, FreeLaneMember, Agent);

crashed(Agent, Victim) generates intClaimCompensation(Victim, Agent);
crashed(Agent, Victim) generates intRequestMerge(CollisionLaneMember,
    FreeLaneMember) if mergePairing(CollisionLaneMember, FreeLaneMember, Agent);
```

```
% initially
initially perm(intClaimCompensation(Victim, Agent)), pow(intClaimCompensation(
    Victim, Agent));
initially perm(intRequestMerge(CollisionLaneMember, FreeLaneMember)), pow(
    intRequestMerge(CollisionLaneMember, FreeLaneMember));
```

---

## A.2 Emergency services institution

Listing A.2: Emergency services institution definition

```
% institution with responsibility for police, fire, ambulance
institution emergencyServices;

% types
type Agent;
type Victim;
type InsuranceState;
type CrimeType;
type PoliceUnit;
type CollisionLaneMember;
type FreeLaneMember;

% events
exogenous event crimeReport(Agent, CrimeType);
exogenous event dispatchUnit(Agent, PoliceUnit);
exogenous event deadline;

%%% obligations
obligation fluent obl(dispatchUnit(Agent, PoliceUnit), deadline, vioDispatchUnit(
    Agent, PoliceUnit));

%%% violation event
violation event vioDispatchUnit(Agent, PoliceUnit);

% fluents
fluent crimeCommitted(Agent, CrimeType);

crimeReport(Agent, CrimeType) initiates crimeCommitted(Agent, CrimeType);
obl(dispatchUnit(Agent, policeUnit1), deadline, vioDispatchUnit(Agent, policeUnit1
    )) when crimeCommitted(Agent, noInsurance);

% initially
initially perm(deadline);
initially perm(crimeReport(Agent, CrimeType));
```

---

## A.3 Insurer institution

Listing A.3: Insurer institution definition

```
% insurance institution
institution insurer;

% types
type Agent;
type Victim;
type InsuranceState;
type CrimeType;
type PoliceUnit;
type CollisionLaneMember;
type FreeLaneMember;

% events
exogenous event checkInsurance(Agent);
exogenous event pay(Victim);
exogenous event deadline;

inst event intCrimeCommitted(Agent, CrimeType);

%%% obligations
obligation fluent obl(pay(Victim), deadline, vioPay(Victim));

%%% violation event
violation event vioPay(Victim);

% fluents
fluent carInsuranceState(Agent, InsuranceState);
fluent payee(Victim, Agent);

obl(pay(Victim), deadline, vioPay(Victim)) when payee(Victim, Agent);
checkInsurance(Agent) generates intCrimeCommitted(Agent, noInsurance) if
    carInsuranceState(Agent, invalid);

% initially
initially perm(deadline);
initially perm(checkInsurance(Agent));
initially perm(intCrimeCommitted(Agent, CrimeType)), pow(intCrimeCommitted(Agent,
    CrimeType));
initially carInsuranceState(centralMember1, invalid);
```

---

## A.4 Motorway management institution

Listing A.4: Motorway management institution definition

```
% motorway management institution
institution motorwaymanagement;

% types
type Agent;
type Victim;
type InsuranceState;
type CrimeType;
type PoliceUnit;
type CollisionLaneMember;
type FreeLaneMember;

% events
exogenous event merge(FreeLaneMember, CollisionLaneMember);
exogenous event deadline;

inst event intPerformMerge(CollisionLaneMember, FreeLaneMember);

%%% obligations
obligation fluent obl(merge(FreeLaneMember, CollisionLaneMember), deadline,
    vioMerge(FreeLaneMember));

% fluents
fluent mergePairing(CollisionLaneMember, FreeLaneMember, Agent);

obl(merge(FreeLaneMember, CollisionLaneMember), deadline, vioMerge(FreeLaneMember))
    when mergePairing(CollisionLaneMember, FreeLaneMember, Agent);
intPerformMerge(CollisionLaneMember, FreeLaneMember) initiates obl(merge(
    FreeLaneMember, CollisionLaneMember), deadline, vioMerge(FreeLaneMember));

%% violations
violation event vioMerge(FreeLaneMember);
```

---



# Bibliography

- E.C. Adam. Fighter cockpits of the future. In *Digital Avionics Systems Conference, 1993. 12th DASC., AIAA/IEEE*, pages 318–323, Oct 1993. doi: 10.1109/DASC.1993.283529. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=283529>. 22
- UK Highways Agency. Traffic flow database system. Accessible via <https://trads.hatris.co.uk>. accessed 26th Jan 2014. 96
- Khaldoun Albarazi, Utayba Mohammad, and Nizar Al-holou. Doppler shift impact on vehicular ad-hoc networks. *Canadian Journal on Multimedia and Wireless Networks*, 2(3):46–56, 2011. 47
- Natasha Alechina, Rafael H Bordini, Jomi Fred Hübner, Mark Jago, and Brian Logan. Belief revision for agentspeak agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1288–1290. ACM, 2006. 179
- Marwa Altayeb and Imad Mahgoub. A survey of vehicular ad hoc networks routing protocols. *International Journal of Innovation and Applied Studies*, 3(3):829–846, 2013. 46
- John R. Anderson. *How Can the Human Mind Occur in the Physical Universe?* OUP USA, 2007. 32
- Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic, and Rudi Studer. A rule-based language for complex event processing and reasoning. In Pascal Hitzler and Thomas Lukasiewicz, editors, *RR*, volume 6333 of *Lecture Notes in Computer Science*, pages 42–57. Springer, 2010. ISBN 978-3-642-15917-6. 28
- Kevin Ashton. That internet of things thing. *RFiD Journal*, 22:97–114, 2009. 31
- Vincent Baines and Julian Padget. A situational awareness approach to intelligent vehicle agents. In Michael Behrisch and Melanie Weber, editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 77–103. Springer International Publishing, 2015. ISBN 978-3-319-15023-9. doi: 10.1007/978-3-319-15024-6\_6. URL [http://dx.doi.org/10.1007/978-3-319-15024-6\\_6](http://dx.doi.org/10.1007/978-3-319-15024-6_6). 20
- Tina Balke, Marina De Vos, and Julian A. Padget. Analysing energy-incentivized cooperation in next generation mobile networks using normative frameworks and an agent-based simulation.

- Future Generation Comp. Syst.*, 27(8):1092–1102, 2011. URL <http://dblp.uni-trier.de/db/journals/fgcs/fgcs27.html#BalkeVP11>. 39, 189
- M Balmer, K Meister, M Rieser, K Nagel, Kay W Axhausen, Kay W Axhausen, and Kay W Axhausen. *Agent-based simulation of travel demand: Structure and computational performance of MATSim-T*. ETH, Eidgenössische Technische Hochschule Zürich, IVT Institut für Verkehrsplanung und Transportsysteme, 2008. 51, 95
- D. Barbieri, D. Braga, S. Ceri, E.D. Valle, Yi Huang, V. Tresp, A. Rettinger, and H. Wermser. Deductive and inductive stream reasoning for semantic social media analytics. *Intelligent Systems, IEEE*, 25(6):32–41, nov.-dec. 2010. ISSN 1541-1672. doi: 10.1109/MIS.2010.142. 28
- Tristan M. Behrens, Koen V. Hindriks, and Jürgen Dix. Towards an environment interface standard for agent platforms. *Ann. Math. Artif. Intell.*, 61(4):261–295, 2011. doi: 10.1007/s10472-010-9215-9. URL <http://dx.doi.org/10.1007/s10472-010-9215-9>. 42
- C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson. Challenges of platooning on public motorways. In *17th World Congress on Intelligent Transport Systems*, 2010. URL <http://www.sartre-project.eu/en/publications/Documents/ITSWCchallengesofplatooningconceptandmodelling10b.pdf>. 46
- Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001. 30
- Bradley J Best, Christian Lebiere, and K Christine Scarpinato. Modeling synthetic opponents in mout training simulations using the act-r cognitive architecture. In *Proceedings of the 11th Conference on Computer Generated Forces and Behavior Representation*, volume 7, 2002. 33
- J. Bezos. 2013 letter to shareholders. <http://phx.corporate-ir.net/External.File?item=UGFyZW50SUQ9MjI4Njc2fENoaWxkSUQ9LTF8VHlwZT0z&t=1>, April 2014. URL <http://phx.corporate-ir.net/External.File?item=UGFyZW50SUQ9MjI4Njc2fENoaWxkSUQ9LTF8VHlwZT0z&t=1>. accessed April 12th, 2014. 95
- Katrin Bilstrup, Annette Böhm, Kristoffer Lidström, Magnus Jonsson, Tony Larsson, Lars Strandén, and Hossein Zakizadeh. Vehicle alert system. In *Proceedings of 14th world congress on intelligent transport system (ITS)*, pages 2–9, 2007. URL <http://www.diva-portal.org/smash/get/diva2:239391/FULLTEXT01.pdf>. 131
- Adrian Boeing and Thomas Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 281–288, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-912-8. doi: 10.1145/1321261.1321312. URL <http://www.adrianboeing.com/pal/papers/p281-boeing.pdf>. 49

- Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761, 2013. 48
- Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 55–60, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-189-7. doi: 10.1145/1410012.1410025. URL <http://doi.acm.org/10.1145/1410012.1410025>. 93
- R. H. Bordini, J. F. Hbner, and M Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley, 2007. 37, 78
- John R Boyd. The essence of winning and losing. *Unpublished lecture notes*, 1996. URL [http://tobeortodo.com/wp-content/uploads/2011/11/essence\\_of\\_winning\\_losing.pdf](http://tobeortodo.com/wp-content/uploads/2011/11/essence_of_winning_losing.pdf). 36
- Jeffrey M. Bradshaw, Paul J. Feltovich, Hyuckchul Jung, Shriniwas Kulkarni, William Taysom, and Andrzej Uszok. Dimensions of adjustable autonomy and mixed-initiative interaction. *Computational Autonomy*, 2969:17–39, 2004. URL <http://www.jeffreybradshaw.net/publications/Dimensionschapter.pdf>. 40
- M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988. URL <http://www.sci.brooklyn.cuny.edu/~sklar/teaching/f08/mas/papers/bratman-israel-pollack-1988-clearer.pdf>. 12, 33
- J. Campos, M. Lpez-Snchez, and M. Esteva. A case-based reasoning approach for norm adaptation. In *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS'10)*, pages 168–176, San Sebastian, Spain, 2010. Springer, Springer. URL <http://www.maia.ub.es/~maite/papers/cbr-hais10-to-publish.pdf>. 40
- J. Chen. Distributed coordination of autonomous agents by communicating on a need-to-know basis. Master of engineering, Massachusetts Institute of Technology, 2003. URL [http://groups.csail.mit.edu/mers/theses/Thesis1.3\\_Judy.pdf](http://groups.csail.mit.edu/mers/theses/Thesis1.3_Judy.pdf). 29
- Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. In Katsumi Inoue, Ken Satoh, and Francesca Toni, editors, *CLIMA VII*, volume 4371 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2006. ISBN 978-3-540-69618-6. 39, 97
- G. Court. Validating the NEC benefits chain. In *Proceedings of the 11th ICCRTS - Coalition Command and Control in the Networked Era*, September 2006. URL [http://www.dodccrp.org/events/11th\\_ICCRTS/html/papers/155.pdf](http://www.dodccrp.org/events/11th_ICCRTS/html/papers/155.pdf). 31

- J. Craighead, R. Murphy, J. Burke, and B. Goldiez. A survey of commercial open source unmanned vehicle simulators. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 852–857, April 2007. doi: 10.1109/ROBOT.2007.363092. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.1143&rep=rep1&type=pdf>. 49
- Stephen Cranefield and Surangika Ranathunga. Embedding agents in business applications using enterprise integration patterns. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS '13, pages 1223–1224, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5. URL <http://dl.acm.org/citation.cfm?id=2484920.2485153>. 42
- Adam Szydłowski Dariusz Pierzchała, Michał Dyk. Distributed military simulation augmented by computational collective intelligence. In *Computational Collective Intelligence. Technologies and Applications*, volume 6922 of *LNCIS*, pages 399–408. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23934-2. doi: 10.1007/978-3-642-23935-9\_39. URL [http://dx.doi.org/10.1007/978-3-642-23935-9\\_39](http://dx.doi.org/10.1007/978-3-642-23935-9_39). 41
- J. Davis. Say hello to Stanley. <http://www.wired.com/wired/archive/14.01/stanley.html>, January 2006. URL [http://www.wired.com/wired/archive/14.01/stanley.html?pg=3&topic=stanley&topic\\_set=](http://www.wired.com/wired/archive/14.01/stanley.html?pg=3&topic=stanley&topic_set=). accessed October 8th, 2011. 44
- Frank Dignum, Cyril Brom, Koen V. Hindriks, Martin D. Beer, and Deborah Richards, editors. *Cognitive Agents for Virtual Environments - First International Workshop, CAVE 2012, Held at AAMAS 2012, Valencia, Spain, June 4, 2012, Revised Selected Papers*, volume 7764 of *Lecture Notes in Computer Science*, 2013. Springer. ISBN 978-3-642-36443-3. 214, 217
- V. Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, 2003. URL <http://igitur-archive.library.uu.nl/dissertations/2003-1218-115420/inhoud.htm>. 38
- F. Dressler, F. Kargl, J. Ott, O. Tonguz, and L. Wischhof. 10402 abstracts collection and executive summary – inter-vehicular communication. In *Inter-Vehicular Communication*, number 10402 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. URL [http://drops.dagstuhl.de/opus/volltexte/2011/2929/pdf/10402\\_abstracts\\_collection.2929.pdf](http://drops.dagstuhl.de/opus/volltexte/2011/2929/pdf/10402_abstracts_collection.2929.pdf). 46
- Michael Düring, Mark Gonter, Falco Thiel, and Florian Weinert. Models enabling simulations of V2X applications regarding emergency vehicles in urban environment. In *SUMO 2014 - Modeling Mobility with Open Data*, volume 24. Deutsches Zentrum für Luft- und Raumfahrt e.V Institut für Verkehrssystemtechnik, May 2014. 131
- Bureau d’Enquêtes et d’Analyses pour la sécurité de l’aviation civile. Final Report On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris. Technical report, Ministère de l’Écologie, du

- Développement durable, des Transports et du Logement, July 2012. URL <http://www.bea.aero/docspa/2009/f-cp090601.en/pdf/f-cp090601.en.pdf>. 24
- Mica Endsley and William M Jones. Situation awareness information dominance and information warfare. Technical report, DTIC Document, 1997. URL <http://www.dtic.mil/dtic/tr/fulltext/u2/a347166.pdf>. 23
- Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995. doi: 10.1518/001872095779049543. URL <http://hfs.sagepub.com/cgi/content/abstract/37/1/32>. 11, 22, 55
- Jakob Engel, Jrgen Sturm, and Daniel Cremers. Camera-based navigation of a low-cost quadcopter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2815–2821, 2012. URL [https://vision.in.tum.de/\\_media/spezial/bib/engell2iros.pdf](https://vision.in.tum.de/_media/spezial/bib/engell2iros.pdf). 93
- R Evertsz, F. Ritter, P. Busetta, M. Pedrotti, and J. Bittner. CoJACK – achieving principled behaviour variation in a moderated cognitive architecture. *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation*, 2008. URL <http://acs.ist.psu.edu/papers/evertszBPRB08.pdf>. 34
- Tim Finin, Jay Weber, Gio Wiederhold, Michael Genesereth, Richard Fritzson, Donald McKay, James McGuire, Richard Pelavin, Stuart Shapiro, and Chris Beck. Draft specification of the kqml agent-communication language, 1993. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.7110&rep=rep1&type=pdf>. 35
- Jannik Fritsch and Sebastian Wrede. An integration framework for developing interactive robots. In *Software Engineering for Experimental Robotics*, pages 291–305. Springer, 2007. URL [https://code.ai.techfak.uni-bielefeld.de/trac/xcf/raw-attachment/wiki/Publications/STAR\\_chapter.pdf](https://code.ai.techfak.uni-bielefeld.de/trac/xcf/raw-attachment/wiki/Publications/STAR_chapter.pdf). 42
- Germano Fronza. Simulador de um ambiente virtual distribuido multiusuario para batalhas de tanques 3D com inteligencia baseada em agentes BDI. <http://campeche.inf.furb.br/tccs/2008-I/2008-1-14-ap-germanofronza.pdf>, July 2008. 37, 48, 88
- Richard M Fujimoto. Time management in the high level architecture. *Simulation*, 71(6):388–400, 1998. URL <http://www.cc.gatech.edu/computing/pads/PAPERS/HLA-PADS96.pdf>. 41
- Martin Gebser, Torsten Grote, Roland Kaminski, Philipp Obermeier, Orkunt Sabuncu, and Torsten Schaub. Stream reasoning with answer set programming: Preliminary report. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *KR*. AAAI Press, 2012. ISBN 978-1-57735-560-1. 28

- Nikki Gordon-Bloomfield. Nissan takes Japanese PM on autonomous LEAF test drive. <http://transportevolved.com/2013/11/11/nissan-takes-japanese-pm-on-autonomous-leaf-test-drive/>, November 2013. URL <http://transportevolved.com/2013/11/11/nissan-takes-japanese-pm-on-autonomous-leaf-test-drive/>. accessed January 19th, 2014. 45
- Ellen Grumert and Andreas Tapani. Microscopic traffic simulation for evaluation of a cooperative variable speed limit system. In *Proceedings of the 1st SUMO User Conference*. Deutsches Zentrum für Luft- und Raumfahrt e.V Institut für Verkehrssystemtechnik, May 2013. 51, 52
- Erico Guizzo. How google’s self-driving car works. <http://spectrum.ieee.org/autoton/robotics/artificial-intelligence/how-google-self-driving-car-works/>, October 2011. URL <http://spectrum.ieee.org/autoton/robotics/artificial-intelligence/how-google-self-driving-car-works/>. accessed May 11th, 2014. 45
- Marcelo T Hama, RS Allgayer, CE Pereira, and RH Bordini. Uavas: Agentspeak agents for unmanned aerial vehicles. In *II Workshop on Autonomous Software Systems, So Paulo, SP, Brazil, Autosoft*, volume 7, 2011. 37
- Garrett Hardin. The tragedy of the commons. *science*, 162(3859):1243–1248, 1968. URL <http://www.sciencemag.org/content/162/3859/1243.full.pdf&embedded=true>. 38, 39, 40
- John Harding, Gregory Powell, Rebecca Yoon, Joshua Fikentscher, Charlene Doyle, Dana Sade, Mike Lukuc, Jim Simons, and Jing Wang. Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application. Technical report, National Highway Traffic Safety Administration, August 2014. URL <http://www.nhtsa.gov/staticfiles/rulemaking/pdf/V2V/Readiness-of-V2V-Technology-for-Application-812014.pdf>. 29
- J. Harrauld and T. Jefferson. Shared situational awareness in emergency management mitigation and response. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 23–23, Jan 2007. doi: 10.1109/HICSS.2007.481. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4076416>. 31
- Nick Hawes and Marc Hanheide. CAST: Middleware for memory-based architectures. In *Proceedings of the AAAI Robotics Workshop: Enabling Intelligence Through Middleware*, July 2010. URL <http://prints.vicos.si/publications/files/59>. 42
- H. Hickey. Move over, Herbie: Stanford team enters autonomous car in driverless race. <http://news.stanford.edu/news/2005/may18/darpasr-051805.html>, May 2005. URL <http://news.stanford.edu/news/2005/may18/darpasr-051805.html>. accessed October 8th, 2011. 44
- Koen V Hindriks, Birna Van Riemsdijk, Tristan Behrens, Rien Korstanje, Nick Kraayenbrink, Wouter Pasma, and Lennard De Rijk. Unreal goal bots. In *Agents for games and*

- simulations II*, pages 1–18. Springer, 2011. URL <http://mmi.tudelft.nl/~birna/publications/2011/hindriks10ags.pdf>. 34
- R. Hourizi. *Awareness beyond mode error*. PhD thesis, University of Bath, 1999. 24
- Nick Howden, Ralph Rönquist, Andrew Hodgson, and Andrew Lucas. Jack intelligent agents—summary of an agent infrastructure. In *5th International conference on autonomous agents*, 2001. 34
- Werner Huber, Michael Lädke, and Rainer Ogger. Extended floating-car data for the acquisition of traffic information. In *Proceedings of the 6th World congress on intelligent transport systems*, pages 1–9, 1999. URL <http://trid.trb.org/view.aspx?id=695348>. 46
- Jomi F Hubner, Jaime S Sichman, and Olivier Boissier. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3):370–395, 2007. URL <http://www.das.ufsc.br/~jomi/pubs/2007/Hubner-ijaose-2007.pdf>. 48
- S. A. Huettel, A. W. Song, and G. McCarthy. Decisions under uncertainty: Probabilistic context influences activation of prefrontal and parietal cortices. *Journal of Neuroscience*, 25(13):3304–3311, 2005. URL <http://www.jneurosci.org/content/25/13/3304.full.pdf>. 25
- ZeroC Inc. The internet communications engine, 2005. URL <http://www.zeroc.com/>. 42
- Daniel Jiang, Qi Chen, and Luca Delgrossi. Optimal data rate selection for vehicle safety communications. In *Proceedings of the Fifth ACM International Workshop on Vehicular Inter-NETworking*, VANET '08, pages 30–38, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-191-0. doi: 10.1145/1410043.1410050. URL <http://doi.acm.org/10.1145/1410043.1410050>. 47
- Randolph M. Jones, John E. Laird, and Paul E. Nielsen. Automated intelligent pilots for combat flight simulation. In Jack Mostow and Chuck Rich, editors, *AAAI/IAAI*, pages 1047–1054. AAAI Press / The MIT Press, 1998. ISBN 0-262-51098-7. 33
- S. Joyeux, R. Alami, and S. Lacroix. A plan manager for multi-robot systems. In *Field and Service Robotics*, volume 42, pages 443–452, 2008. URL <http://hal.archives-ouvertes.fr/docs/00/16/67/98/PDF/roby-2007-fsr.pdf>. 35
- S. Karim and C. Heinze. Experiences with the design and implementation of an agent-based autonomous UAV controller. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 2005. URL [http://pdf.aminer.org/000/056/094/experiences\\_with\\_the\\_design\\_and\\_implementation\\_of\\_an\\_agent\\_based.pdf](http://pdf.aminer.org/000/056/094/experiences_with_the_design_and_implementation_of_an_agent_based.pdf). 36, 37
- D. Kern, M. Mller, S. Schneegaß, L. Wolejko-Wolejszo, and A. Schmidt. CARS - Configurable Automotive Research Simulator. In *Automotive User Interfaces and Interactive Applications*, 2008. URL <http://cars.pcuie.uni-due.de/fileadmin/cars/pdf/CARS.pdf>. 25



- Kyeong Tae Kim. STVC: Secure Traffic-light to Vehicle Communication. In *ICUMT*, pages 96–104. IEEE, 2012. ISBN 978-1-4673-2016-0. URL <http://dblp.uni-trier.de/db/conf/icumt/icumt2012.html#Kim12>. 47
- Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347. ACM, 1997. 47
- Tobias Knerr. Merging elevation raster data and OpenStreetMap vectors for 3D rendering. Master’s thesis, University of Passau, May 2013. URL <http://tobias-knerr.de/publications/thesis/MasterThesisKnerr.pdf>. 118, 119
- A. Komenda, J. Vokrinek, M. Cap, and M. Pechoucek. Developing multiagent algorithms for tactical missions using simulation. *Intelligent Systems, IEEE*, 28(1):42–49, Jan 2013. ISSN 1541-1672. doi: 10.1109/MIS.2012.90. 50
- Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012. URL <http://elib.dlr.de/80483/>. 51, 95
- R. Kusterer. *jMonkeyEngine 3.0 Beginner’s Guide*. Packt Publishing, 2012. ISBN 1849516464, 9781849516464. 88, 108
- M Laine and K Sälä. Performance evaluation of XMPP on the Web. Technical report, Aalto University, April 2012. 77
- John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1 – 64, 1987. ISSN 0004-3702. doi: [http://dx.doi.org/10.1016/0004-3702\(87\)90050-6](http://dx.doi.org/10.1016/0004-3702(87)90050-6). URL <http://www.sciencedirect.com/science/article/pii/0004370287900506>. 32
- JeeHang Lee, Vincent Baines, and Julian Padget. Decoupling cognitive agents and virtual environments. In [Dignum et al. \[2013\]](#), pages 17–36. ISBN 978-3-642-36443-3. URL <http://www.cs.bath.ac.uk/~jl495/papers/cave2012.pdf>. 20, 76, 112
- JeeHang Lee, Tingting Li, and Julian Padget. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013. ISSN 1546-427X. doi: 10.1002/cav.1517. URL <http://onlinelibrary.wiley.com/doi/10.1002/cav.1517/pdf>. 37, 39, 76, 97
- JeeHang Lee, Julian Padget, Brian Logan, Daniela Dybalova, and Natasha Alechina. N-jason: Run-time norm compliance in agentspeak(l). In Fabiano Dalpiaz, Jrgen Dix, and M.Birna van Riemsdijk, editors, *Engineering Multi-Agent Systems*, volume 8758 of *Lecture Notes in Computer Science*, pages 367–387. Springer International Publishing, 2014. ISBN 978-3-319-14483-2. doi: 10.1007/978-3-319-14484-9\_19. URL [http://dx.doi.org/10.1007/978-3-319-14484-9\\_19](http://dx.doi.org/10.1007/978-3-319-14484-9_19). 110



- I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla. On the effectiveness of an opportunistic traffic management system for vehicular networks. In *IEEE Transactions on Intelligent Transportation Systems*. IEEE Intelligent Transportation Systems Society, 2011. URL [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5970119](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5970119). 47
- Tingting Li, Tina Balke, Marina De Vos, Julian Padget, and Ken Satoh. Legal conflict detection in interacting legal systems (JURIX 2013). In Kevin D. Ashley, editor, *Legal Knowledge and Information Systems*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 107–116. IOS Press, 2013a. 98
- Tingting Li, Tina Balke, Marina De Vos, Julian A. Padget, and Ken Satoh. A model-based approach to the automatic revision of secondary legislation. In Enrico Francesconi and Bart Verheij, editors, *ICAIL*, pages 202–206. ACM, 2013b. ISBN 978-1-4503-2080-1. 98
- Yuyi Li, Kai Ying, Peng Cheng, Hui Yu, and Hanwen Luo. Cooperative data dissemination in cellular-vanet heterogeneous wireless networks. In *High Speed Intelligent Communication Forum (HSIC), 2012 4th International*, pages 1–4, May 2012. doi: 10.1109/HSIC.2012.6212991. 47
- Richard Lichtenstein, Daniel Clarence Smith, Jordan Lynne Ambrose, and Laurel Anne Moody. Headphone use and pedestrian injury and death in the united states: 20042011. *Injury Prevention*, 2012. doi: 10.1136/injuryprev-2011-040161. URL <http://injuryprevention.bmj.com/content/early/2012/01/03/injuryprev-2011-040161.abstract>. 57
- L.N. Long, S.D. Hanford, O. Janrathitikarn, G.L. Sinsley, and J.A. Miller. A review of intelligent systems software for autonomous vehicles. In *Computational Intelligence in Security and Defense Applications, 2007. CISDA 2007. IEEE Symposium on*, pages 69–76, April 2007. doi: 10.1109/CISDA.2007.368137. 32, 33
- Martin Lowson. New approach to effective and sustainable urban transport. *Transportation Research Record: Journal of the Transportation Research Board*, 1838(1):42–49, 2003. URL [www.ltrc.lsu.edu/TRB\\_82/TRB2003-000140.pdf](http://www.ltrc.lsu.edu/TRB_82/TRB2003-000140.pdf). 45
- J. Markoff. Google cars drive themselves, in traffic. <http://www.nytimes.com/2010/10/10/science/10google.html>, October 2010. URL <http://www.nytimes.com/2010/10/10/science/10google.html>. accessed October 8th, 2011. 45
- Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. OpenDS: A new open-source driving simulator for research. In *4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 7–8, October 2012. URL <http://www.auto-ui.org/12/docs/AutomotiveUI-2012-Adjunct-Proceedings.pdf>. 50

- Natasha McCarthy. Autonomous systems: Social, legal and ethical issues. *The Royal Academy of Engineering*, 2009. URL [http://www.raeng.org.uk/societygov/engineeringethics/pdf/Autonomous\\_Systems\\_Report\\_09.pdf](http://www.raeng.org.uk/societygov/engineeringethics/pdf/Autonomous_Systems_Report_09.pdf). 11
- Dejan Mitrović, Mirjana Ivanović, and Hans-Dieter Burkhard. Intelligent Jason Agents in Virtual Soccer Simulations. In Matthias Klusch, Matthias Thimm, and Marcin Paprzycki, editors, *Multiagent System Technologies*, volume 8076 of *Lecture Notes in Computer Science*, pages 334–345. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40775-8. doi: 10.1007/978-3-642-40776-5\_28. URL [http://dx.doi.org/10.1007/978-3-642-40776-5\\_28](http://dx.doi.org/10.1007/978-3-642-40776-5_28). 48
- Geoffrey P Morgan, Frank E Ritter, William E Stevenson, Ian N Schenck, and Mark A Cohen. dtank: An environment for architectural comparisons of competitive agents. In *Proceedings of the 14th Conference on Behavior Representation and Modeling Simulation*, pages 133–140, 2005. URL <http://cc.ist.psu.edu/BRIMS2013/archives/2005/papers/05-BRIMS-043.pdf>. 48
- Katia Moskvitch. 'Talking' cars could reduce motorway pile-ups. <http://www.bbc.co.uk/news/technology-14125245>, July 2011. URL <http://www.bbc.co.uk/news/technology-14125245>. accessed October 8th, 2011. 29
- F. C. Müller-Lyer. Optische urteilstäuschungen. archiv für anatomie und physiologie. In *Physiologische Abteilung*, volume 2, pages 263–270, 1889. 27
- Nathan Naylor. U.S. Department of Transportation Announces Decision to Move Forward with Vehicle-to-Vehicle Communication Technology for Light Vehicles. <http://www.nhtsa.gov/About+NHTSA/Press+Releases/2014/USDOT+to+Move+Forward+with+Vehicle-to-Vehicle+Communication+Technology+for+Light+Vehicles>, February 2014. 46
- Emma Norling. Folk psychology for human modelling: Extending the bdi paradigm. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems- Volume 1*, pages 202–209. IEEE Computer Society, 2004. 34
- Emma Jane Norling. *Modelling Human Behaviour with BDI Agents*. PhD thesis, University of Melbourne, 2012. URL [https://minerva-access.unimelb.edu.au/bitstream/handle/11343/37081/281964\\_Norlan.pdf?sequence=1](https://minerva-access.unimelb.edu.au/bitstream/handle/11343/37081/281964_Norlan.pdf?sequence=1). 33
- Douglass C. North. Institutions. *Journal of Economic Perspectives*, 5(1):97–112, 1991. doi: 10.1257/jep.5.1.97. URL <http://www.aeaweb.org/articles.php?doi=10.1257/jep.5.1.97>. 38
- Elinor Ostrom. *Governing the commons: The evolution of institutions for collective action*. Cambridge university press, 1990. 38, 97

- M. Papageorgiou, E. Kosmatopoulos, and I. Papamichail. Effects of variable speed limits on motorway traffic flow. *Transportation Research Record*, 2047(-1):37–48, 2008. URL [http://www.academia.edu/4696023/Effects\\_of\\_Variable\\_Speed\\_Limits\\_on\\_Motorway\\_Traffic\\_Flow](http://www.academia.edu/4696023/Effects_of_Variable_Speed_Limits_on_Motorway_Traffic_Flow). 135
- C. Pinart, J.C. Calvo, L. Nicholson, and J.A Villaverde. Ecall-compliant early crash notification service for portable and nomadic devices. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5, April 2009. doi: 10.1109/VETECS.2009.5073518. 139
- Tomás Plch, Tomas Jedlicka, and Cyril Brom. HLA Proxy: Towards connecting agents to virtual environments by means of High Level Architecture (HLA). In [Dignum et al. \[2013\]](#), pages 1–16. ISBN 978-3-642-36443-3. 41
- Stefan Poslad. Specifying protocols for multi-agent systems interaction. *ACM Trans. Auton. Adapt. Syst.*, 2(4), November 2007. ISSN 1556-4665. doi: 10.1145/1293731.1293735. URL <http://doi.acm.org/10.1145/1293731.1293735>. 35
- S. Ranathunga, S. Cranefield, and M. Purvis. Interfacing a cognitive agent platform with a virtual world: a case study using Second Life (extended abstract),. In Sonenberg Tumer, Yolum and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*,, pages 1181–1182, 2011. URL <http://www.ifaamas.org/Proceedings/aamas2011/papers/B20.pdf>. 37
- Surangika Ranathunga, Stephen Cranefield, and Martin Purvis. Identifying events taking place in second life virtual environments. *Applied Artificial Intelligence*, 26(1-2):137–181, 2012. doi: 10.1080/08839514.2012.629559. URL <http://otago.ourarchive.ac.nz/bitstream/handle/10523/4069/IdentifyingEventsTakingPlaceinSecondLifeVirtualEnvironments.pdf?sequence=1>. 28
- Anand S Rao and Michael P Georgeff. Modeling rational agents within a BDI-architecture. *KR*, 91:473–484, 1991. URL <https://www.cs.drexel.edu/~greenie/cs510/rao91modeling-1.pdf>. 33
- K. Reichel, N. Hochgeschwender, and H. Voos. OpCog: An industrial development approach for cognitive agent systems in military UAV applications (short paper). In Nishiyama (eds.) Berger, Burg, editor, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 97–100, May 2008. URL [http://ifaamas.org/Proceedings/aamas08/proceedings/pdf/industrial\\_application\\_track/AAMAS08\\_IndTrack\\_27.pdf](http://ifaamas.org/Proceedings/aamas08/proceedings/pdf/industrial_application_track/AAMAS08_IndTrack_27.pdf). 36
- Frank E Ritter, Sue E Kase, Damodar Bhandarkar, Bil Lewis, and Mark A Cohen. dTank updated: exploring moderated behavior in a light-weight synthetic environment. In *Proceedings of the 16th Conference on Behavior Representation in Modeling and Simulation*, pages 51–60, 2007. URL <http://ritter.ist.psu.edu/papers/ritterKBLC07.pdf>. 49

- John R Searle. *The construction of social reality*. Simon and Schuster, 1995. 97
- Giuseppe P Settembre, Paul Scerri, Alessandro Farinelli, Katia Sycara, and Daniele Nardi. A decentralized approach to cooperative situation assessment in multi-robot systems. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems- Volume 1*, pages 31–38. International Foundation for Autonomous Agents and Multiagent Systems, 2008. 37
- A. Shendarkar, K. Vasudevan, and Y. Son S. Lee. Crowd simulation for emergency response using BDI agents based on immersive virtual reality. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*. IEEE, 2008. URL <http://www.informs-sim.org/wsc06papers/067.pdf>. 34
- Guilherme Soares, Jose Macedo, Zafeiris Kokkinogenis, and Rosaldo Rossetti. An integrated framework for Multi-Agent Traffic Simulation using SUMO and JADE. In *Proceedings of the 1st SUMO User Conference*. Deutsches Zentrum für Luft- und Raumfahrt e.V Institut für Verkehrssystemtechnik, May 2013. 51, 52
- L. Spear and V. Baines. An initial conceptual model for morale factors. In *Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation*,, pages 31–38, March 2009. URL [http://brimsconference.org/archives/2009/papers/BRIMS2009\\_009.pdf](http://brimsconference.org/archives/2009/papers/BRIMS2009_009.pdf). 185
- Mehdi Fallah Tafti. An investigation on the approaches and methods used for Variable Speed Limit control. In *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*, 2008. URL <http://ictte.ir/8/286.pdf>. 135, 136
- Nassim Nicholas Taleb. *Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets*. Random House, London, 2 edition, 2007. 56
- Tarik Taleb, Abderrahim Benslimane, and Khaled Ben Letaief. Toward an effective risk-conscious and collaborative vehicular collision avoidance system. *Vehicular Technology, IEEE Transactions on*, 59(3):1474–1486, 2010. 136
- Ali Murat Tiryaki, Sibel Öztuna, Oguz Dikenelli, and Riza Cenk Erdur. SUNIT: A unit testing framework for test driven development of multi-agent systems. In Lin Padgham and Franco Zambonelli, editors, *Agent-Oriented Software Engineering VII*, volume 4405 of *Lecture Notes in Computer Science*, pages 156–173. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-70944-2. doi: 10.1007/978-3-540-70945-9\_10. URL [http://dx.doi.org/10.1007/978-3-540-70945-9\\_10](http://dx.doi.org/10.1007/978-3-540-70945-9_10). 114
- C. Urmson. The latest chapter for the self-driving car: mastering city street driving. <http://googleblog.blogspot.co.uk/2014/04/the-latest-chapter-for-self-driving-car.html>,

- April 2014. URL <http://googleblog.blogspot.co.uk/2014/04/the-latest-chapter-for-self-driving-car.html>. accessed May 12th, 2014. 28, 45
- C. Urmson and W. Whittaker. Self-driving cars and the urban challenge. *IEEE Intelligent Systems*, 23:66–68, March 2008. URL <http://dl.acm.org/citation.cfm?id=1399090.1399250>. 45
- Chris Urmson, J Andrew Bagnell, Christopher R Baker, Martial Hebert, Alonzo Kelly, Raj Rajkumar, Paul E Rybski, Sebastian Scherer, Reid Simmons, Sanjiv Singh, et al. Tartan racing: A multi-modal approach to the DARPA urban challenge. Technical report, Carnegie Mellon University, 2007. URL [http://www.ri.cmu.edu/pub\\_files/2007/4/Tartan\\_Racing.pdf](http://www.ri.cmu.edu/pub_files/2007/4/Tartan_Racing.pdf). 44
- Various. *BRITISH ALMANAC OF THE SOCIETY FOR THE DIFFUSION OF USEFUL KNOWLEDGE FOR THE YEAR 1866*. Charles Knight & Co., 1866. 56
- Gediminas Vilkas. Parliament supports life-saving eCall system in cars. [http://www.europarl.europa.eu/pdfs/news/expert/infopress/20140224IPR36860/20140224IPR36860\\_en.pdf](http://www.europarl.europa.eu/pdfs/news/expert/infopress/20140224IPR36860/20140224IPR36860_en.pdf), February 2014. URL [http://www.europarl.europa.eu/pdfs/news/expert/infopress/20140224IPR36860/20140224IPR36860\\_en.pdf](http://www.europarl.europa.eu/pdfs/news/expert/infopress/20140224IPR36860/20140224IPR36860_en.pdf). accessed July 19th, 2014. 139
- Jiri Vokrinek, Martin Schaefer, and Daniele Pinotti. Multi-agent traffic simulation for human-in-the-loop cooperative drive systems testing. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1691–1692. International Foundation for Autonomous Agents and Multiagent Systems, 2014. URL <http://aamas2014.lip6.fr/proceedings/aamas/pl691.pdf>. 50
- Volkswagen. Audi in the simTD large-scale test study: the ”traffic light info online” project. [http://www.volkswagenag.com/content/vwcorp/info\\_center/en/news/2013/06/audi\\_simTD.html](http://www.volkswagenag.com/content/vwcorp/info_center/en/news/2013/06/audi_simTD.html), June 2013. URL [http://www.volkswagenag.com/content/vwcorp/info\\_center/en/news/2013/06/audi\\_simTD.html](http://www.volkswagenag.com/content/vwcorp/info_center/en/news/2013/06/audi_simTD.html). accessed January 19th, 2014. 47, 126
- Rashid A Waraich, David Charypar, Michael Balmer, Kay W Axhausen, Rashid A Waraich, Rashid A Waraich, Kay W Axhausen, and Kay W Axhausen. *Performance improvements for large scale traffic simulation in matsim*. ETH, Eidgenössische Technische Hochschule Zürich, IVT, Institut für Verkehrsplanung und Transportsysteme, 2009. 51
- Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007. URL <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>. 93
- M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems*. MIT, 1999. 77

Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10:115–152, 6 1995. ISSN 1469-8005. doi: 10.1017/S0269888900008122. URL <http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/ker95.pdf>. 34

Stephane Maag Xiaoping Che. Testing protocols in internet of things by a formal passive technique. *SCIENCE CHINA Information Sciences*, 57(3):32101, 2014. doi: 10.1007/s11432-014-5068-x. URL [http://info.scichina.com:8084/sciFe/EN/abstract/article\\_513730.shtml](http://info.scichina.com:8084/sciFe/EN/abstract/article_513730.shtml). 76

XMPP Standards Foundation. The XMPP standards foundation homepage. Retrieved from <http://www.xmpp.org>, 20130129, no date. URL <http://www.xmpp.org>. 74